(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10247183 A**

(43) Date of publication of application: **14.09.98**

(51) Int. Cl **G06F 15/16**
G06F 9/46
// G06F 13/00

(21) Application number: **09346401**

(22) Date of filing: **16.12.97**

(30) Priority: **23.12.96 US 96 780015**

(71) Applicant: **INTERNATL BUSINESS MACH CORP <IBM>**

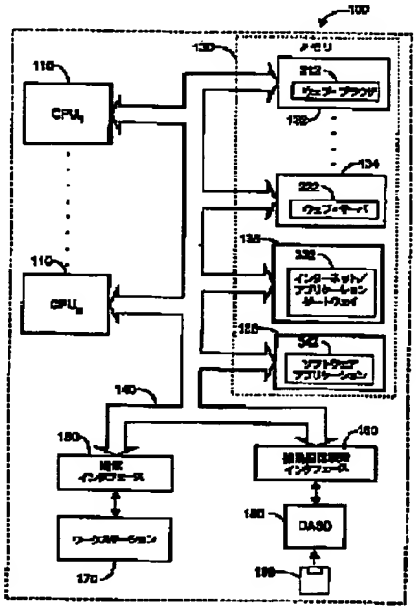
(72) Inventor: **MARSHA LYNN BRANT**
KENNETH EDGAR BROWN
PARNELL JAMES DYKES
ERIC DWAIN LINDBERGH
DIANE IREIN OLSON
JEFFERY EDWARD SELDEN
DEVON DANIEL SNYDER
JAMES ORIN WALTZ

(54) **COMPUTER SYSTEM AND METHOD**

(57) Abstract:

PROBLEM TO BE SOLVED: To easily access many different application programs in a WWW via a standardized GUI by transmitting and receiving the data to and from plural web browsers and performing the communication between these browsers and a software application.

SOLUTION: The data are inputted via a web browser 212, and these input data are communicated to a web server application 222. The application 222 transfers the proper input data to an application gateway 332 to authenticate and also uniquely identify the browser 212. The gateway 332 decides whether the user of the browser 212 is permitted to access a software application 342. Then the gateway 332 can access the application 342.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平10－247183

(43)公開日　平成10年(1998)９月14日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	15/16	３７０	Ｇ０６Ｆ	15/16	３７０Ｎ
	9/46	３６０		9/46	３６０Ｂ
// Ｇ０６Ｆ	13/00	３５４		13/00	３５４Ｚ

審査請求　未請求　請求項の数27　ＯＬ　（全 43 頁）

(21)出願番号　特願平9－346401

(22)出願日　平成９年(1997)12月16日

(31)優先権主張番号　０８／７８００１５

(32)優先日　1996年12月23日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

(72)発明者　マーシャ・リン・プラント
アメリカ合衆国55901　ミネソタ州ロチェスターフォーティーサード・ストリート　ノースウェスト　1902

(74)代理人　弁理士　坂口　博　（外１名）

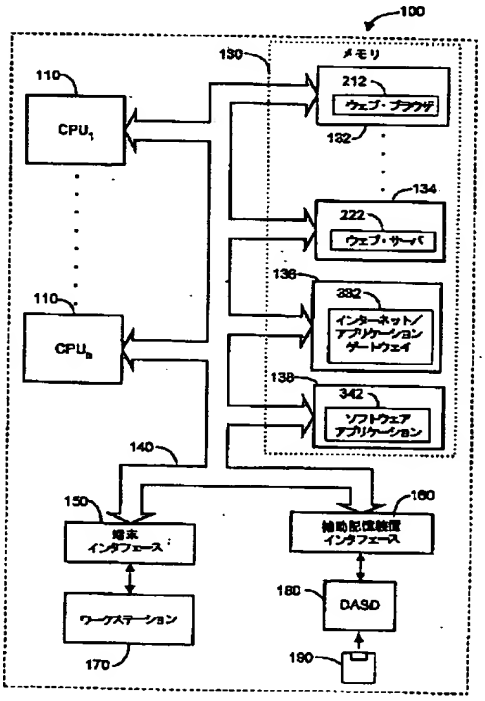
最終頁に続く

(54)【発明の名称】　コンピュータシステム及び方法

(57)【要約】

【課題】　本発明の目的は、WWW上で共通ユーザ・インターフェースを介して多数の異なるアプリケーション・プログラムに簡単にアクセスする能力を提供することである。

【解決手段】　WWW上でソフトウェア・アプリケーションにアクセスするための標準的な手順、ルーチン、ツール及びソフトウェア「フック」を提供することによって、ソフトウェア開発者が、アプリケーション・プログラムの機能性に努力を集中し、ＨＴＭＬを使用してアプリケーション・プログラム用のＧＵＩインターフェースを簡単に提供できるようになる。



1

【特許請求の範囲】

【請求項１】少なくとも１つの中央処理装置（ＣＰＵ）と、

ＣＰＵに結合されたメモリと、

メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、共通ユーザ・インターフェースを介して複数のウェブ・ブラウザへまたはこれらからデータを送受する能力を有し、データの識別及び追跡のために識別子機構を使用する、トランザクション・サポート機構とを含む、ワールド・ワイド・ウェブ上で複数のウェブ・ブラウザとソフトウェア・アプリケーションとの間で通信するための共通ユーザ・インターフェースを提供するコンピュータ・システム。

【請求項２】さらに、セキュリティ機構を含み、セキュリティ機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、セキュリティ機構が、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間に結合され、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間のインターフェースを提供し、セキュリティ機構が、複数のウェブ・ブラウザからユーザ入力を受け取り、セキュリティ機構が、受け取った入力に対応するソフトウェア・アプリケーションの認証パラメータを取り出す、請求項１のコンピュータ・システム。

【請求項３】さらに、インターフェース機構を含み、インターフェース機構が、少なくとも１つの変数を処理するためのゲートウェイ機構を含み、ゲートウェイ機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ゲートウェイ機構が、ソフトウェア・アプリケーションの再プログラミングを必要としない、複数のウェブ・ブラウザとソフトウェア・アプリケーションとの間の通信のための汎用共通ゲートウェイ・インターフェースを含む、請求項１のコンピュータ・システム。

【請求項４】さらに、切断機構を含み、切断機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、切断機構が、ソフトウェア・アプリケーション・プロセスが再開される時にデータを取り出すことができるように、ソフトウェア・アプリケーション・プロセスが中止される時に、複数のウェブ・ブラウザのうちの１つとソフトウェア・アプリケーション・プロセスとの間の会話のそれぞれに関連する状態データ及び会話識別子を記憶する、請求項１のコンピュータ・システム。

【請求項５】メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間に結合され、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間のインターフェースを提供し、複数のウェブ・ブラウザからユーザ入力を受け取り、受け取った入力に対応するソフトウェア・アプリケーションの認証パラメータを取り出す、セキュリティ機構と、

2

少なくとも１つの変数を処理するためのゲートウェイ機構を含み、ゲートウェイ機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ゲートウェイ機構が、ソフトウェア・アプリケーションの再プログラミングを必要としない、複数のウェブ・ブラウザとソフトウェア・アプリケーションとの間の通信のための汎用共通ゲートウェイ・インターフェースを含む、インターフェース機構と、

メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ソフトウェア・アプリケーション・プロセスが再開される時にデータを取り出すことができるように、ソフトウェア・アプリケーション・プロセスが中止される時に、複数のウェブ・ブラウザのうちの１つとソフトウェア・アプリケーション・プロセスとの間の会話のそれぞれに関連する状態データ及び会話識別子を記憶する、切断機構とをさらに含む、請求項１のコンピュータ・システム。

【請求項６】トランザクション・サポート機構が、さらに、ソフトウェア・アプリケーションへのネイティブ・インターフェースと通信するための機構を含む、請求項１のコンピュータ・システム。

【請求項７】トランザクション・サポート機構が、ウェブ・サーバ・アプリケーション及びソフトウェア・アプリケーションと通信するアプリケーション・ゲートウェイを含み、アプリケーション・ゲートウェイが、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行され、アプリケーション・ゲートウェイが、識別子機構を含み、識別子機構が、複数のウェブ・ブラウザのそれぞれのために識別子を生成し、ソフトウェア・アプリケーションからのデータを、複数のウェブ・ブラウザのうちの識別子に対応する選択された１つに経路指定する、請求項１のコンピュータ・システム。

【請求項８】アプリケーション・ゲートウェイが、複数のウェブ・サーバから受け取るデータを処理し、アプリケーション・プログラムから受け取るデータを処理する、請求項７のコンピュータ・システム。

【請求項９】ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、請求項１のコンピュータ・システム。

【請求項１０】さらに、ソフトウェア・アプリケーションの指示の下で実行される少なくとも１つのアクティビティ・プログラムと通信する少なくとも１つのアクティビティ・プログラム・インターフェース（ＡＰＩ）を含み、少なくとも１つのアクティビティ・プログラム・インターフェースが、少なくとも１つのアクティビティ・プログラムとアプリケーション・ゲートウェイとの間で通信する、請求項１のコンピュータ・システム。

【請求項１１】複数の中央処理装置（ＣＰＵ）と、

複数のＣＰＵに結合されたメモリと、

それぞれがメモリ内に常駐し、複数のＣＰＵのうちの少

なくとも１つによって実行される、複数のウェブ・ブラウザと、
複数のウェブ・ブラウザのうちの少なくとも１つと通信する、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ウェブ・サーバ・アプリケーションと、
メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ソフトウェア・アプリケーションと、
ウェブ・サーバ・アプリケーション及びソフトウェア・アプリケーションへのネイティブ・インターフェースと通信する、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、アプリケーション・ゲートウェイとを含み、アプリケーション・ゲートウェイが、複数のウェブ・ブラウザのそれぞれのために識別子を生成し、ソフトウェア・アプリケーションからのデータを、複数のウェブ・ブラウザのうちの識別子に対応する選択された１つに経路指定する、識別子機構を含むワールド・ワイド・ウェブ上でウェブ・ブラウザとソフトウェア・アプリケーションとの間で通信するための共通ユーザ・インターフェースを提供するコンピュータ・システム。
【請求項１２】アプリケーション・ゲートウェイが、ウェブ・サーバ・アプリケーション及びアプリケーション・プログラムから受け取るデータを処理する、請求項１１のコンピュータ・システム。
【請求項１３】ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、請求項１１のコンピュータ・システム。
【請求項１４】さらに、ソフトウェア・アプリケーションの指示の下で実行される少なくとも１つのアクティビティ・プログラムと通信する少なくとも１つのアクティビティ・プログラム・インターフェース（ＡＰＩ）を含み、少なくとも１つのアクティビティ・プログラム・インターフェースが、少なくとも１つのアクティビティ・プログラムとアプリケーション・ゲートウェイとの間で通信する、請求項１１のコンピュータ・システム。
【請求項１５】ウェブ・サーバ・アプリケーションが、複数のウェブ・ブラウザのうちの１つから渡された認証データから、選択されたウェブ・ブラウザがウェブ・サーバ・アプリケーションへのアクセスを許可されるかどうかを判定する認証機構を含み、
ウェブ・サーバが、複数のウェブ・ブラウザから受け取るデータ及びアプリケーション・ゲートウェイから受け取るデータを処理する請求項１１のコンピュータ・システム。
【請求項１６】ウェブ・ブラウザが、複数のＣＰＵのうちの少なくとも１つによってクライアント・ワークステーション上で実行される、請求項１１のコンピュータ・システム。

【請求項１７】ウェブ・サーバ・アプリケーションが、複数のＣＰＵのうちの少なくとも１つによって、ウェブ・サーバ・コンピュータ上で実行される、請求項１１のコンピュータ・システム。
【請求項１８】アプリケーション・ゲートウェイが、複数のＣＰＵのうちの少なくとも１つによって、ウェブ・サーバ・コンピュータ上で実行される、請求項１１のコンピュータ・システム。
【請求項１９】アプリケーション・ゲートウェイが、複数のＣＰＵのうちの少なくとも１つによって、第１のコンピュータ上で実行される、請求項１１のコンピュータ・システム。
【請求項２０】ソフトウェア・アプリケーションが、複数のＣＰＵのうちの少なくとも１つによって、第２のコンピュータ上で実行される、請求項１１のコンピュータ・システム。
【請求項２１】アプリケーション・ゲートウェイが、複数のＣＰＵのうちの少なくとも１つによって、第２のコンピュータ上で実行される、請求項１１のコンピュータ・システム。
【請求項２２】複数の中央処理装置（ＣＰＵ）を提供するステップと、
複数のＣＰＵに結合されたメモリを提供するステップと、
複数のＣＰＵのうちの少なくとも１つによって、メモリ内に常駐する複数のウェブ・ブラウザのうちの少なくとも１つを実行するステップと、
メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ウェブ・サーバ・アプリケーションを提供するステップと、
メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ソフトウェア・アプリケーションを提供するステップと、
メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、アプリケーション・ゲートウェイを提供するステップと、
認証データ及び環境データをウェブ・サーバ・アプリケーションに送ることによって、複数のウェブ・ブラウザのうちの選択された１つが、ソフトウェア・アプリケーションへのアクセスを開始するステップと、
認証データが選択されたウェブ・ブラウザにウェブ・サーバ・アプリケーションへのアクセスを許可する場合に、環境データを処理するステップと、
処理された環境データをアプリケーション・ゲートウェイに出力するステップと、
選択されたウェブ・ブラウザと、ソフトウェア・アプリケーションによって実行される所望のプロセスとに対応する識別子を生成するステップと、
メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ウェブ・ブラウザからユーザ入力を受け取り、

受け取った入力に対応するソフトウェア・アプリケーションの認証パラメータを取り出す、セキュリティ機構を提供するステップと、
メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ウェブ・ブラウザとソフトウェア・アプリケーションとの間で変数及びテンプレートを送受する、インターフェース機構を提供するステップと、
メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、会話が再開されてソフトウェア・アプリケーションによって所望のプロセスが実行される時に状態データを取り出すことができるように、会話が中止される時にウェブ・ブラウザとソフトウェア・アプリケーションとの間の会話に関連する状態及び会話識別子を記憶する、切断機構を提供するステップと、
所望のプロセスを実行した結果を、識別子を有するアプリケーション・ゲートウェイに返すステップと、
識別子に基づいて、複数のブラウザのうちのどれに結果を送らなければならないかを決定するステップと、
アプリケーション・ゲートウェイからウェブ・サーバ・アプリケーションに結果を送るステップと、
識別子に対応する選択された１つのウェブ・ブラウザに、サーバから結果を送るステップとを含む、ワールド・ワイド・ウェブ上でウェブ・ブラウザとソフトウェア・アプリケーションとの間で通信するための共通ユーザ・インターフェースを提供するための、コンピュータ実施される方法。
【請求項２３】ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、請求項２２の方法。
【請求項２４】ウェブ・ブラウザを走行させるクライアント・ワークステーションと、
ウェブ・サーバ・アプリケーションを走行させるウェブ・サーバ・コンピュータと、
アプリケーション・ゲートウェイを走行させる第１コンピュータと、
ソフトウェア・アプリケーションを走行させる第２コンピュータと、
ウェブ・ブラウザとウェブ・サーバ・アプリケーションとの間でデータを伝送できるようにする、ウェブ・ブラウザとウェブ・サーバ・アプリケーションとの間の通信機構と、
ウェブ・サーバ・アプリケーションとアプリケーション・ゲートウェイとの間でデータを伝送できるようにする、ウェブ・サーバ・アプリケーションとアプリケーション・ゲートウェイとの間の通信機構と、
アプリケーション・ゲートウェイとソフトウェア・アプリケーションとの間でデータを伝送できるようにする、アプリケーション・ゲートウェイとソフトウェア・アプリケーションとの間の通信機構と、
ウェブ・ブラウザとソフトウェア・アプリケーションとの間で変数及びテンプレートを送受する、インターフェース機構と、
ウェブ・ブラウザとソフトウェア・アプリケーションとの間に結合され、ウェブ・ブラウザとソフトウェア・アプリケーションとの間のインターフェースを提供する、セキュリティ機構と、
会話が再開されてソフトウェア・アプリケーションによって所望のプロセスが実行される時に状態データを取り出すことができるように、会話が中止される時にウェブ・ブラウザとソフトウェア・アプリケーションとの間の会話に関連する状態及び会話識別子を記憶する、切断機構と、
ウェブ・ブラウザがワールド・ワイド・ウェブ上でソフトウェア・アプリケーションと通信できるようにする、複数のアプリケーション・プログラミング・インターフェースとを含む、ワールド・ワイド・ウェブ上でウェブ・ブラウザとソフトウェア・アプリケーションの間で通信するための共通ユーザ・インターフェースを提供するためのシステム。
【請求項２５】ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、請求項２４のシステム。
【請求項２６】ウェブ・サーバ・コンピュータが、第１コンピュータを含む、請求項２４のシステム。
【請求項２７】第１コンピュータが、第２コンピュータを含む、請求項２４のシステム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、全般的にはワールド・ワイド・ウェブ上での対話に関し、具体的には、ワールド・ワイド・ウェブを介してソフトウェア・アプリケーションへのアクセスを提供するための方法及び装置に関する。
【０００２】
【従来の技術】しばしば、１９４８年のＥＤＶＡＣコンピュータ・システムの開発が、コンピュータ時代の始まりとして引用される。それ以降、コンピュータ・システムは、極度に洗練された装置に発展し、コンピュータ・システムは、多くの異なる設定で見出される可能性がある。コンピュータ・システムには、通常、ハードウェア（たとえば半導体、回路基板など）とソフトウェア（たとえばコンピュータ・プログラム）の組み合わせが含まれる。半導体加工とコンピュータ・アーキテクチャの進歩によってコンピュータ・ハードウェアの性能が高まるにつれて、ハードウェアの高性能を利用するためにより洗練されたコンピュータ・ソフトウェアが発展し、１～２年前に存在したシステムよりはるかに強力な現在のコンピュータ・システムがもたらされた。
【０００３】コンピュータ・システムには、通常、コンピュータの基本機能を制御するオペレーティング・シス

テム・ソフトウェアと、オペレーティング・システムの制御下で走行して所望のタスクを実行する１つまたは複数のソフトウェア・アプリケーションが含まれる。たとえば、通常のIBM Personal ComputerではＯＳ／２オペレーティング・システムが走行し、ＯＳ／２オペレーティング・システムの制御下で、ユーザは、ワード・プロセッサなどのアプリケーション・プログラムを実行することができる。コンピュータ・システムの能力が高まるにつれて、高性能コンピュータ・システム用に設計されたソフトウェア・アプリケーションは、極度に強力になってきた。

【０００４】技術面の他の変化も、コンピュータの使い方に大きく影響してきた。たとえば、コンピュータの広範囲の普及によって、コンピュータが互いに通信できるようになるコンピュータ・ネットワークの開発が促進された。パーソナル・コンピュータ（ＰＣ）の導入に伴って、多数の人々がコンピューティングを利用できるようになった。パーソナル・コンピュータ用のネットワークが開発されて、個々のユーザが互いに通信できるようになった。この形で、１企業内の多数の人間が、単一のコンピュータ・システム上で走行するソフトウェア・アプリケーションを用いて、ネットワークを介して同時に通信できるようになった。

【０００５】最近非常に人気のある重要なコンピュータ・ネットワークがインターネットである。インターネットは、現代のコンピュータ及びネットワークの普及から発生し、全体として「ワールド・ワイド・ウェブ」またはWWWを構成するウェブ・ページによって互いにリンクされたコンピュータ・システムの洗練された国際ネットワークに発展した。WWWへのアクセスを望む個々のＰＣ（すなわちワークステーション）のユーザは、通常は、ウェブ・ブラウザと称するソフトウェア・アプリケーションを介してこれを行う。ウェブ・ブラウザは、ウェブ・サーバと称する他のコンピュータへのWWWを介する接続を行い、ユーザのワークステーションに表示される情報をウェブ・サーバから受信する。ユーザに表示される情報は、通常は、ハイパーテキスト・マークアップ言語（ＨＴＭＬ）と称する特殊な言語を使用して構成される。ＨＴＭＬを使用するウェブ・ブラウザは、現在、市販されているコンピュータ・システムのほとんどすべてで使用可能であり、コンピュータとモデムにアクセスできる人であれば事実上誰でもWWWにアクセスできるようになっている。WWWは、ますます一般的になりつつあるが、WWWにアクセスするコンピュータ・ユーザの急激な増加が、それに付随する問題をもたらした。これらの問題のいくつかを、以下で示す。

【０００６】インターネットとWWWの人気が高まるのに伴って、インターネットが売上と効率の両方を高めるための新しい方法をもたらすことが認識された。ウェブ・ブラウザを有するユーザが、会社のソフトウェア・アプリケーションと直接対話できるならば、所与のトランザクションが単純化される。たとえば、誰かが普通にレンタ・カーを予約する方法を検討する。その人は、レンタ・カー代理店に電話し、電話を介して自分の情報（すなわち、氏名、住所、クレジット・カード番号など）をレンタ・カー代理人に知らせる。代理人は、この情報を自動車レンタル・ソフトウェア・アプリケーションに入力して、自動車を予約するプロセスを初期設定しなければならない。ウェブ・ユーザ用のより効率的な自動車予約システムでは、ユーザが自動車レンタル・ソフトウェア・アプリケーションと直接対話できるようになるはずである。これによって、現在レンタ・カー代理人が実行している作業の大半が除去されるはずである。しかし、ウェブ・ユーザと直接対話できる自動車レンタル・ソフトウェア・アプリケーションを案出するには、カスタム・インターフェース・ソフトウェアを作成する必要がある。同様に、WWWを介してアクセスされる異なるソフトウェア・アプリケーションごとに、カスタム・ユーザ・インターフェースを作成しなければならない。好ましいインターフェースは、グラフィカル・ユーザ・インターフェース（ＧＵＩ）になるはずである。あるソフトウェア・アプリケーション用のカスタムＧＵＩを生成する処理は、時間がかかり、高価であり、通常は、他のソフトウェア・アプリケーションとの通信には使用できない独自ユーザ・インターフェースがもたらされる。これは、会社がWWWを介するソフトウェア・アプリケーションへのアクセスを提供しなくなる大きな阻害要因となることを意味する。

【０００７】さらに、多数のコンピュータ・ユーザは、ハードウェアまたはソフトウェア的に非常に異なる形態のコンピュータ・プラットフォームを採用している。たとえば、ＩＢＭ互換パーソナル・コンピュータは、現在市販されているもっとも一般的なタイプのコンピュータであるが、他社は、現在導入され使用されている非常に多数のコンピュータ・システムを有する非常に異なる製品系列を開発してきた。これらの全く異なるハードウェア・システムでは、通常は完全に異なるオペレーティング・システムが使用される。これらのさまざまなハードウェア・システム及びソフトウェア・システムの存在は、通常は、異なるハードウェア・プラットフォームのそれぞれに所与のソフトウェア・アプリケーション用のカスタムＧＵＩを「移植」または変換するために、完全に新規のプログラミングと開発の労力を必要とする。多くの会社は、最も一般的なハードウェアとソフトウェアの組み合わせだけをサポートし、これによって、市場のシェアが制限され、そのソフトウェア・アプリケーションにアクセスできるユーザの数が少なくなる。

【０００８】複数の関連しないユーザ・インターフェースの問題によって、やはりWWWを介するソフトウェア・アプリケーションのすばやい採用を阻止する可能性が

あるもう１つの問題が明らかになる。ある会社が、所与のソフトウェア・アプリケーション用のカスタムＧＵＩを開発するコストを吸収したと仮定しても、１つの会社によって開発されたそのＧＵＩは、別の会社によってその会社のソフトウェア・アプリケーション用に開発されたＧＵＩとは非常に異なる可能性が高い。通常、各ソフトウェア・ベンダは、特定のソフトウェア・アプリケーションごとにカスタムＧＵＩを作成し、その結果、ユーザは、ＷＷＷを介してソフトウェア・アプリケーションにアクセスする時には必ず、そのソフトウェア・アプリケーションに固有の特徴にアクセスできるようになる。しかし、各製品は異なる特徴を有し、各ベンダはアプリケーション／ユーザ対話に関して異なる標準を有するので、その結果、ユーザが出会うソフトウェア・アプリケーションのそれぞれに非常に異なるユーザ・インターフェースが存在することがしばしばである。ユーザは、アクセスしたいソフトウェア・アプリケーションのそれぞれと対話するための基本的なスキルを「再学習」しなければならないことがしばしばである。

【０００９】ＷＷＷ上でのソフトウェア・アクセスに関する現在の状況は、パーソナル・コンピュータの初期の時代に多少似ている。１９８０年代初期に、ＩＢＭパーソナル・コンピュータ（ＰＣ）が導入され、業界標準のハードウェア・プラットフォームとしてすばやく採用された。しかし、ハードウェア・プラットフォームは相対的に標準的であったが、独立ソフトウェア・ベンダのそれぞれが、それぞれのアプリケーション・プログラムのために大きく異なるユーザ・インターフェースを作成した。これが、パーソナル・コンピュータのユーザによる新しいアプリケーション・プログラムのすばやい採用の妨げになった。ユーザは、長い訓練なしに新しいソフトウェア・アプリケーションを効率的に使用することができなかったので、多数の新しいソフトウェア・アプリケーションが採用されなかった。

【００１０】

【発明が解決しようとする課題】しかし、現在、新しい標準がパーソナル・コンピュータ・ソフトウェア・アプリケーション用に開発され、採用されており、これによって、新しいソフトウェア・アプリケーションのために行わなければならない再学習の量が劇的に減少した。ＩＢＭ社のＯＳ／２などの最新技術のＧＵＩオペレーティング・システムでは、共通のユーザ動作の多くが標準化され、独立ソフトウェア開発者及びベンダに、最も標準的なユーザ・インターフェースの構成要素及び特徴にアクセスするのに必要な「フック」またはプログラミング・ツールが提供され、これによって、エンド・ユーザの学習曲線が劇的に低くなった。残念ながら、この標準化の努力は、まだＷＷＷには浸透していない。実際、ソフトウェア・アプリケーションの配布システムとしてＷＷＷを簡単に利用できることが、現在ＷＷＷ上で入手できる全く異なるソフトウェアの急激な増加をつのらせている。さらに、入手可能な最も強力なソフトウェア・アプリケーションのうちの一部は、直観的でもグラフィカルでもないユーザ・インターフェースを有する。新しいアプリケーションごとに新しいインターフェースを再学習する過程は、冗長で時間がかかり、非生産的になる傾向を有する。したがって、現在多数のソフトウェア・アプリケーションがＷＷＷ上でアクセス可能ではあるが、これらのさまざまなプログラムにアクセスする方法の学習は、エンド・ユーザにとって、時間がかかり、いらいらさせられ、威嚇的になる可能性がある。

【００１１】

【課題を解決するための手段】ＷＷＷ上でソフトウェア・アプリケーションへのアクセスを提供することの重要性と、既存の解決策の現在の制約の両方を認識して、本発明は、ＷＷＷ上で標準化されたＧＵＩを介して多数の異なるアプリケーション・プログラムに簡単にアクセスする能力を提供する。ＷＷＷ上でソフトウェア・アプリケーションにアクセスするための標準的な手順、ルーチン、ツール及びソフトウェア「フック」を提供することによって、ソフトウェア開発者が、アプリケーション・プログラムの機能性に努力を集中し、ＨＴＭＬを使用してアプリケーション・プログラム用のＧＵＩインターフェースを簡単に提供できるようになる。

【００１２】上で述べたように、ＨＴＭＬは、現在市販されているコンピュータ・システムのほとんどすべてで使用できる周知の言語である。さらに、ＨＴＭＬは、かなり明確に制御され、標準化された言語なので、新しいソフトウェア・アプリケーション機能は、それが開発され、ＨＴＭＬによってサポートされた時に追加できる。さらに、ＨＴＭＬは、広範囲に採用された非独自技術なので、本発明によって、非常に小規模のソフトウェア開発者にも、巨大な市場へのオープン・アクセスを提供できる。さらに、本発明を用いると、ソフトウェア開発者は、標準アクセス・プロトコルを採用でき、これによって、ＨＴＭＬを認識するブラウザを利用できるコンピュータ・システムのすべてに対するサポートを提供できるようになる。最後に、ユーザ・インターフェース、認証／セキュリティ及びウェブ・トランザクション・サポートの問題に対する、実施しやすく標準化された解決を提供することによって、本発明の共通ユーザ・インターフェースは、以前の解決に存在する限界を克服する。

【００１３】本発明の前述及び他の特徴及び長所は、添付図面に図示された、本発明の好ましい実施例の以下の詳細な説明から明白になる。

【００１４】

【発明の実施の形態】

概要

ウェブ・トランザクション

ここで図２を参照すると、クライアント・ワークステー

ション２１０で走行する標準的なウェブ・ブラウザ２１２と、ウェブ・サーバ・コンピュータ・システム２２０で走行するウェブ・サーバ・アプリケーション２２２の間の通常のトランザクションは、接続２１６を介して発生する。クライアント・ワークステーション２１０は、ローカル・エリア・ネットワーク（ＬＡＮ）を介するか他のタイプのコンピュータ・ネットワークまたは他の相互接続を介して、他のコンピュータ・システムに結合することができる。同様に、ウェブ・サーバ・コンピュータ・システム２２０も、他のコンピュータに結合することができる。クライアント・ワークステーション２１０は、ウェブ・ブラウザ２１２を使用することによってＷＷＷへのアクセスを提供できるコンピュータであれば何でもよい。これには、ハンドヘルド・コンピュータ、ポータブル・コンピュータまたはラップトップ・コンピュータ、標準的なデスクトップ・コンピュータ・システム、パーソナル・ディジタル・アシスタント（ＰＤＡ）、メインフレームに接続された非プログラム式端末などが含まれる。

【００１５】ウェブ・ブラウザ２１２は、クライアント・ワークステーション２１０のユーザが接続２１６を介して他のコンピュータと通信できるようにする、クライアント・ワークステーション２１０で走行するソフトウェア・プログラムである。ウェブ・ブラウザ２１２には、ＷＷＷ上でデータを送受信する能力を有するウェブ・ブラウザのすべてが含まれる。これには、IBM社のWeb Explorer、Netscape社のNavigator、Microsoft社のInternet Explorer、Apple Computer社のCyberDogなどの市販ソフトウェア・アプリケーションと、ＷＷＷ上の情報のアクセスまたは処理のための、現在存在するか将来に開発される他のソフトウェア・アプリケーションが含まれる。接続２１６の好ましい実施態様は、インターネットへの適当な接続であり、これには、ハードワイヤ接続、モデムまたは高速Ｔ１回線を介する電話アクセス、赤外線その他の無線通信、コンピュータ・ネットワーク通信（有線または無線）、現在既知であるか将来に開発される他の適当なコンピュータ間接続が含まれる。

【００１６】クライアント・ワークステーション２１０とウェブ・サーバ・コンピュータ・システム２２０は、物理的または論理的に同一のコンピュータ・システムとすることができることに留意されたい。ウェブ・ブラウザ２１２は、通常は、クライアント・ワークステーション２１０のユーザにＨＴＭＬデータのページを表示する。他のタイプ（ＨＴＭＬ以外）のデータも、ウェブ・ブラウザ２１２に送信される可能性があり、これには、テキスト・データ、グラフィカル・データ（たとえばＧＩＦ（Graphic Image Format）ファイル）、オーディオ・データまたはサウンド・ファイル（たとえばＷＡＶファイル）、ジャバ・アプレット（実行可能コード）、ＭＩＭＥ（Multipurpose Internet Mail Extensions）データ（前述及び他のデータ型の組み合わせが含まれる可能性がある）と称する特殊なデータが含まれる。

【００１７】ウェブ・サーバ・アプリケーション２２２は、クライアント・ワークステーション２１０のユーザがウェブ・サーバ・コンピュータ・システム２２０によって制御される情報にアクセスできるようにする、ウェブ・サーバ・コンピュータ・システム２２０で走行するソフトウェア・プログラムである。本発明によるウェブ・サーバ・アプリケーション２２２の好ましい実施態様の１つは、IBM社のInternet Connection Serverなどの市販ウェブ・サーバ・アプリケーションである。他のアプリケーションも、本発明との互換性を有する。ウェブ・サーバ・コンピュータ・システム２２０は、通常は、クライアント・ワークステーション２１０でユーザが行った動作を反映する、ウェブ・ブラウザ２１２による要求に応答して、ウェブ・ブラウザ２１２にＨＴＭＬデータのページを出力する。さらに、上で説明したように、ウェブ・サーバ・コンピュータ・システム２２０は、ウェブ・ブラウザ２１２に他のタイプのデータを出力することもできる。出力データには、静的ＨＴＭＬページ（ページの内容が変化しないことを意味する）または、動的に決定され、出力データに挿入されなければならないデータを含めることができる。ウェブ・サーバ・アプリケーション２２２は、ウェブ・サーバ・コンピュータ・システム２２０内のメモリからまたは他のコンピュータ・システムから受け取る部分から出力データ（たとえばＨＴＭＬページ）を動的に作成することができ、また、以前にまたは別のコンピュータによって開発されたＨＴＭＬページまたは他の情報を単純に渡すことができる。

【００１８】ウェブ・ブラウザ２１２は、通常は、接続２１６を介してウェブ・サーバ・コンピュータ・システム２２０に入力（たとえばＵＲＬ（Uniform Resource Locator）やＨＴＭＬページ）を送信することによって、ウェブ・サーバ・アプリケーション２２２と対話する。この入力は、通常はハイパーテキスト転送プロトコル（ＨＴＴＰ）１．０を使用して送信される。ウェブ・サーバ・アプリケーション２２２が走行するウェブ・サーバ・コンピュータ・システム２２０は、ウェブ・ブラウザ２１２から入力を受信し、これに応答して、ウェブ・ブラウザ２１２にデータ（たとえばＨＴＭＬページ）を出力する。上で述べた処理は、インターネットを介する基本的なトランザクションを示すものであり、本発明の範囲に含まれる多数の詳細及び変形は、本発明の概念を理解するための簡単な文脈を提供するために本明細書では開示されないことを理解されたい。

【００１９】ウェブ・サーバ・コンピュータ・システム２２０は、コモン・ゲートウェイ・インターフェース（ＣＧＩ）モジュールを含む、他の多数のソフトウェア構成要素も有する可能性がある。ＣＧＩモジュールは、

ウェブ・サーバ・アプリケーション２２２と他のソフトウェア・アプリケーションの間のインターフェースとして使用することができる。たとえば、ＣＧＩモジュールは、ウェブ・サーバ・アプリケーション２２２とカレンダ・ソフトウェア・アプリケーションの間のリンクを提供でき、これによって、ウェブ・サーバ・アプリケーション２２２が、たとえば、ウェブ・ブラウザ２１２に出力するウェブ・ページに動的なカレンダ情報を挿入できるようになる。したがって、ＣＧＩを用いると、ウェブ・サーバが、他のソフトウェア・アプリケーションからの動的データを配布できるようになる。残念ながら、ＣＧＩのプログラム作成は、ＣＧＩが満足しなければならない仕様が多数あるので、時間のかかる作業である。さらに、一般に、ウェブ・サーバ・アプリケーション２２２とインターフェースする異なるソフトウェア・アプリケーションごとに、別々のＣＧＩが必要になる。さらに、異なるユーザに同一のソフトウェア・アプリケーションまたはデータへの異なるレベルのアクセス権を与えるなど、異なる機能を実行するためには、異なるＣＧＩが必要になる可能性がある。一般に、ＣＧＩの数が増えるにつれて、ウェブ・サーバ・コンピュータ・システム２２０の性能が低下する。

【００２０】ウェブ・ページ

ここで図３を参照すると、ウェブ・ページは、主に、クライアント・ワークステーション２１０のモニタでの表示を目的とするビジュアル・データである。ウェブ・ページは、一般に、ハイパーテキスト・マークアップ言語（ＨＴＭＬ）で記述される。ウェブ・サーバ・コンピュータ・システム２２０で走行するウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２からのウェブ・ページ要求を受信した時に、ＨＴＭＬでウェブ・ページを作成するか、事前に作成されたウェブ・ページを含むファイルを取り出し、接続２１６を介して要求元のウェブ・ブラウザ２１２に送信する。ウェブ・ブラウザ２１２は、ＨＴＭＬを理解し、解釈し、そのウェブ・ページをクライアント・ワークステーション２１０のモニタに出力する。この結果の、ユーザの画面に表示されるウェブ・ページには、テキスト、グラフィックス及びリンク（他のウェブ・ページのＵＲＬアドレス）が含まれる可能性がある。これらの他のウェブ・ページ（すなわち、リンクによって表されるウェブ・ページ）は、同一のウェブ・サーバ上にあっても、異なるウェブ・サーバ上にあってもよい。ユーザは、マウスまたは他のポインティング装置を使用してこれらのリンクをクリックすることによって、これらの他のウェブ・ページを取り出すことができる。全世界の他のサーバ上の他のウェブ・ページへのリンクを有するウェブ・ページのシステム全体が、集合的に「ワールド・ワイド・ウェブ」（ＷＷＷ）を構成する。

【００２１】一部のウェブ・ページは、ウェブ・ブラウザ２１２からの入力を誘うように設計されている。たとえば、ウェブ・ページのＨＴＭＬフォームでユーザの氏名を要求でき、ＨＴＭＬラジオ・ボタンを使用して特定の機能を選択するようにユーザに要求できる。これらの要求は、ウェブ・サーバ・アプリケーション２２２からウェブ・ブラウザ２１２に送信される。ウェブ・ユーザは、要求された情報を入力し、ウェブ・サーバ・アプリケーション２２２にそのページの実行要求をもう一度送ることによって応答し、ウェブ・サーバ・アプリケーション２２２は、ユーザから受信した入力データを解析する。残念ながら、ユーザは、入力の要求に即座に応答しない場合がある。実際、ユーザが、要求された入力を発行する前に数分、数時間または数日待つ可能性があり、その代わりに、トランザクションを完了せずに切断する可能性もあることが予想される。この時間の間、ウェブ・サーバ・アプリケーション２２２は、通常は、さまざまなプロセスが走行したままの状態で、予期される応答を待ちながらその入力を待つ。これは、入力を待つ間に走行中のプロセスが活動状態のままにされ、ウェブ・サーバ・アプリケーション２２２の資源が非効率的に独占されるという問題を引き起こす可能性がある。

【００２２】ＡＰＩ

アプリケーション・プログラミング・インターフェース（ＡＰＩ）は、所与のソフトウェア・アプリケーションの特定の機能へのアクセスを提供するために、プログラム開発者によって使用される。各アプリケーション・プログラムは、第三者が特定の特徴にアクセスでき、そのアプリケーション・プログラムを他のプログラムとインターフェースでき、エンド・ユーザにアクセスを提供できるようにするＡＰＩを有する。各アプリケーション・プログラムは、通常は独自のＡＰＩを有するが、異なるベンダによって提供される異なるＡＰＩの機能性と使用は、かなり似通っていることがしばしばである。多数の異なるソフトウェア・アプリケーションに共通して見られるＡＰＩの例を、図３に示されるコンピュータ・システム３００で走行するソフトウェア・アプリケーション３４２に関して下で提示する。

【００２３】ＳＥＮＤ　ＡＰＩは、ウェブ・ブラウザ２１２にデータ（ＨＴＭＬページか他のタイプの形で）を送信するために発行される。ソフトウェア・アプリケーション３４２は、ＳＥＮＤ　ＡＰＩを呼び出し、ＳＥＮＤ　ＡＰＩは、送信要求を送信し、インターネット／アプリケーション・ゲートウェイ（以下ではゲートウェイと呼称する）３３２とウェブ・サーバ・アプリケーション２２２を介して適当なウェブ・ブラウザ２１２にデータを送信する。

【００２４】ＲＥＣＥＩＶＥ　ＡＰＩは、ウェブ・ブラウザ２１２からデータを受信するために発行される。これには、必要に応じてＣＧＩデータと環境データを含めることができる。ソフトウェア・アプリケーション３４

２は、ＲＥＣＥＩＶＥ　ＡＰＩを呼び出して、ゲートウェイ３３２に入力を待つように指示する。入力が到着したならば、ゲートウェイ３３２は、そのデータを処理のためソフトウェア・アプリケーション３４２に渡す。

【００２５】ウェブ・セキュリティ

この１～２年にWWWが爆発的に増大したので、ウェブ・セキュリティに関する関心がますます高まっている。具体的に言うと、ウェブ・ブラウザとウェブ・サーバが、他のコンピュータ資源（すなわち、ソフトウェア・アプリケーション、データ・ファイル、ＨＴＭＬウェブ・ページなど）へのウェブ・ベースのアクセスを提供するのに使用されている場合に、これらの資源を安全に保たなければならない。これには、これらの資源へのアクセスが、認可されたウェブ・ユーザだけに許可されることを保証することが含まれる。

【００２６】場合によっては、異なるユーザが、ウェブ・サーバを介してアクセス可能な異なる資源へのアクセスを許可されるシステムを提供しなければならない。たとえば、あるウェブ・サーバが、２つのソフトウェア・アプリケーションへのウェブ・アクセスを提供し、各ソフトウェア・アプリケーションが、複数のデータベースへのアクセスを提供する場合がある。ユーザの中には、一方のソフトウェア・アプリケーションへのアクセスは必要だが、他方へのアクセスが不要なユーザも、両方のソフトウェア・アプリケーションへのアクセスが必要だが、これらのソフトウェア・アプリケーションによって制御される指定されたデータベースへのアクセスだけが必要なユーザもいる。あるユーザが、２つのソフトウェア・アプリケーションのうちの一方へのアクセスを許可され、セキュリティ検査が、ウェブ・サーバ・レベルだけで行われる場合、そのユーザは、第２のソフトウェア・アプリケーションへのアクセスを許可されていないにもかかわらず、ユーザへのアクセス許可によって、両方のソフトウェア・アプリケーションへのアクセスが許可される。各ソフトウェア・アプリケーションへのアクセスを許可する前にセキュリティ検査が実行される場合であっても、アクセス許可によって、ユーザは、ソフトウェア・アプリケーションによってアクセス可能なデータベースのすべてにアクセスできるようになる。したがって、承認されたユーザのアクセス権を維持しながら、承認されないユーザが重要な資源へのアクセス権を得ることができないことを保証するための、より洗練されたセキュリティ検査技法が必要である。

【００２７】通常のウェブ・セキュリティでは、特定のウェブ・サーバへまたはそのウェブ・サーバを介する特定の資源へのアクセスのために特定のウェブ・ユーザを認証するのに、パスワードとユーザＩＤの組み合わせが使用される。ウェブ・ユーザは、このような保護された資源へのアクセスを試みる時には、ユーザＩＤとパスワードを供給しなければならない。これは、通常、ウェブ・ユーザがユーザＩＤとパスワードを入力することをウェブ・サーバに要求させ、このユーザＩＤとパスワードをウェブ・ブラウザが記憶し、検証または認証のためウェブ・サーバ・アプリケーションに送り返すことによって達成される。

【００２８】この認証処理は、通常は、ウェブ・サーバ・アプリケーションを介してアクセス可能な特定の資源のそれぞれについて繰り返される。したがって、複数の保護資源にアクセスするウェブ・ユーザは、資源ごとにパスワードとユーザＩＤを入力するよう促され、そうすることが必要になる。たとえば、ユーザは、ウェブ・サーバへのアクセス権を得るために自分のパスワードとユーザＩＤを入力しなければならず、そのウェブ・サーバを介するソフトウェア・アプリケーションへのアクセス権を得るために第２のパスワードとユーザＩＤを入力しなければならず、特定のソフトウェア・アプリケーション・データベースへのアクセス権を得るために第３のパスワードとユーザＩＤを入力しなければならない場合がある。この場合、ユーザは、パスワードとユーザＩＤを何度も発行する手間に加えて、多数のパスワードとユーザＩＤを記憶する必要がある。パスワードとユーザＩＤがどのレベルでも同一の場合であっても、同一の情報を何度も繰り返して入力するのは単調で退屈になる。

【００２９】従来のウェブ・サーバ認証システムでは、ウェブ・サーバに実行依頼を送信するたびに、ウェブ・ブラウザがパスワードとユーザＩＤを再送信する。したがって、パスワードとユーザＩＤは、「スヌーブ」（すなわち、ウェブ・サーバとウェブ・ブラウザの間の伝送の、許可されず望ましくない傍受）の危険に繰り返しさらされる。一部のシステムは、ユーザがパスワードとユーザＩＤを変更することを定期的に要求することによってこの問題を制限しようとするが、複数のソフトウェア・アプリケーションにアクセスするのに複数のパスワードと複数のユーザＩＤが必要になる場合には、これが大きな問題になる可能性がある。

【００３０】従来技術の解決に関連する問題のために、一部のシステム操作員は、追加のセキュリティ保護手段を除去し、ユーザＩＤとパスワードを用いるウェブ・サーバ認証だけに頼るようになっている。やはり、この解決を採用できるのは、ウェブからアクセス可能な資源への無許可アクセスの危険性が高くても許容される場合に限られる。

【００３１】詳細な説明

本発明による、WWW上でウェブ・ブラウザから複数のソフトウェア・ベンダによって提供される複数のソフトウェア・アプリケーションにグラフィカルな共通ユーザ・インターフェースを提供するための装置及び方法を開示する。このシステムには、ウェブ・ブラウザ、ウェブ・サーバ・アプリケーション、アプリケーション・ゲートウェイ及び少なくとも１つのソフトウェア・アプリケ

ーションを実行する１つまたは複数のコンピュータが含まれる。本発明のシステム及び方法を用いると、ウェブ・ブラウザのユーザが、共通ユーザ・インターフェースを使用して複数のソフトウェア・アプリケーションにアクセスできるようになる。ユーザは、ウェブ・ブラウザを介してデータを入力し、このデータが、ウェブ・サーバ・アプリケーションに通信される。ウェブ・サーバ・アプリケーションは、ウェブ・ブラウザを認証し、ユーザの要求を一意に識別し、追跡するためのデータを含む適当な入力データを、アプリケーション・ゲートウェイに渡す。アプリケーション・ゲートウェイは、ソフトウェア・アプリケーションに対する適当なコマンドを書式化することによって、要求に対する応答を簡単にする。ソフトウェア・アプリケーションは、適当なデータをアプリケーション・ゲートウェイに出力することによって応答する。この出力データには、出力データとその出力データを要求した特定のウェブ・ブラウザの突合せに使用することができる識別子が含まれる。このシステムは、多数のウェブ・ブラウザに、多数のソフトウェア・アプリケーションへの同時アクセスも提供する。さらに、本発明を用いると、標準的なWWW　HTMLブラウザのいずれもが、サーバに対する真にプラットフォーム独立のクライアントとして動作できるようになり、すべてのサーバ・サブシステムが対話に用いる一貫した効率的なエンド・ユーザ・インターフェースが提供される。

【００３２】理解しやすいＧＵＩインターフェースに統合された時に、本発明の好ましい実施例を構成する主要な構成要素は、ＭＩＭＥ（Multipurpose Internet Mail Extension）データ形式を受信し、単一のクライアント・アプリケーションに送信できるWWWトランザクション・サポート技術を提供するカスタム・アプリケーション・プログラミング・インターフェース（ＡＰＩ）と、サーバ側セキュリティのためのサポートを提供すると同時に、エンド・ユーザをセキュリティ・システムの複雑さから保護することができるセキュリティ機構と、通常のアプリケーション処理の入出力要件の大半に対処できるＨＴＭＬテンプレート／フォームなどの共通WWWユーザ・インターフェース構成要素を定義する機構と、切断、処理タイムアウト、クライアントまたはサーバの異常終了、他の通常の処理の問題などの複雑な問題の処理を自動化する切断処理機構の４つである。この４つのサブシステムのそれぞれを、以下の節で詳細に説明する。

【００３３】ここで図１を参照すると、本発明の好ましい実施例によるコンピュータ・システムには、複数の中央処理装置（ＣＰＵ）１１０、端末インターフェース１５０、補助記憶装置インターフェース１６０、ワークステーション１７０、直接アクセス記憶装置（ＤＡＳＤ）１８０、フロッピ・ディスク１９０、バス１４０及び、さまざまなソフトウェア・プログラムを格納するための複数の記憶位置を含むメモリ１３０が含まれる。この例では、メモリ１３０に、記憶位置１３２で走行するウェブ・ブラウザ２１２、記憶位置１３４で走行するウェブ・サーバ・アプリケーション２２２、記憶位置１３６で走行するゲートウェイ３３２及び記憶位置１３８で走行するソフトウェア・アプリケーション３４２が含まれる。

【００３４】ＣＰＵ１１０は、システム１００の計算機能と制御機能を実行する。システム１００に関連するＣＰＵのすべてに、マイクロプロセッサなどの単一の集積回路をそれぞれ個別に含めるか、中央処理装置の機能を達成するため共同して働く適当な個数の集積回路デバイスまたは回路基板を含めることができる。すべてのＣＰＵが、メモリ１３０内に格納されたプログラムを適当に実行する能力を有し、これらのプログラムまたはシステム１００内で発生し得る他の活動に応答して動作する能力を有する。

【００３５】メモリ１３０は、当業者に既知のいずれかのタイプのメモリである。これには、ダイナミック・ランダム・アクセス・メモリ（ＤＲＡＭ）、スタティックＲＡＭ（ＳＲＡＭ）、フラッシュ・メモリ、キャッシュ・メモリなどが含まれる。図１には明示されていないが、メモリ１３０は、単一のタイプのメモリ構成要素とするか、多数の異なるタイプのメモリ構成要素から構成することができる。たとえば、記憶位置１３２で走行するウェブ・ブラウザ２１２は、システム１００のキャッシュ・メモリの一部とすることができる。さらに、メモリ１３０とＣＰＵ１１０は、集団としてシステム１００を構成する複数の異なるコンピュータの間で分散することができる。たとえば、ウェブ・ブラウザ２１２が、CPU_1を有するコンピュータに常駐し、ウェブ・サーバ・アプリケーション２２２が、別のCPU_2を有するもう１つのコンピュータ・システムに常駐し、ゲートウェイ３３２が、異なるCPU_{n-1}を有する第３のコンピュータ・システムに常駐し、ソフトウェア・アプリケーション３４２が、異なるCPU_nを有する第４のコンピュータに常駐することができる。図１のシステム１００は、ＣＰＵ１１０の物理的な位置またはメモリ１３０内の記憶位置に関する制限なしに、本発明の顕著な特徴の多くを図示するに過ぎない。

【００３６】バス１４０は、システム１００のさまざまな構成要素の間でプログラム、データ、状況及び他の形の情報または信号を伝送するために働く。バス１４０の好ましい実施例は、当業者に既知の、コンピュータ・システム及び構成要素を接続する適当な物理的手段または論理的手段のいずれかである。これには、直接ハードワイヤ接続、インターネット接続、イントラネット接続、光ファイバ、赤外線（ＩＲ）及び他の形の無線接続が含まれるが、これらに制限されるものではない。コンピュータ・システム及び構成要素を接続するための多数の代

替の方法及び材料を、本発明と共に使用するために簡単に適合できることが予期される。これには、現在既知ではなく、将来に開発される方法及び材料が含まれる。

【００３７】端末インターフェース１５０を用いると、人間のユーザが、通常は、プログラム式であるワークステーション１７０を介してシステム１００と通信できるようになる。図１に示されたシステム１００には、単一のワークステーション１７０だけが含まれるが、実際にシステム１００に接続されるワークステーションの数は、システムの設計及びユーザの選択の関数であることを理解されたい。ワークステーション１７０は、人間がシステム１００と対話できるようにする、ダム端末または他の非プログラム式コンピュータ入出力装置とすることもできる。

【００３８】補助記憶装置インターフェース１６０は、当業者に既知の、記憶装置をコンピュータ・システムにインターフェースする方法のいずれかを表す。補助記憶装置インターフェース１６０を用いると、ＤＡＳＤ１８０などの補助記憶装置を、システム１００の他の構成要素に接続できるようになり、これと通信できるようになる。１つの補助記憶装置インターフェース１６０だけが図示されているが、本発明では、複数のインターフェースとＤＡＳＤ１８０などの複数の補助記憶装置が予想されている。図１に示されているように、ＤＡＳＤ１８０は、フロッピ・ディスク１９０上のプログラムまたはデータを読み書きできるフロッピ・ディスク駆動装置とすることができる。ＤＡＳＤ１８０も、当業者に既知の他のいずれかのタイプのＤＡＳＤとすることができる。これには、ＣＤ－ＲＯＭ駆動装置、ハード・ディスク装置、光ディスク装置などが含まれる。フロッピ・ディスク１９０は、当業者に既知の通常の３．５インチ磁気媒体ディスクを表す。

【００３９】ここで図５を参照すると、本発明の好ましい実施例によるＧＵＩを介してＷＷＷ上のソフトウェア・アプリケーションにアクセスするための方法５００が示されている。ＷＷＷを介するソフトウェア・アプリケーションへのアクセスを開始するために、ユーザは、ウェブ・ブラウザからウェブ・サーバへの入力を生成する何らかの動作を行う（ステップ５１０）。ウェブ・サーバは、ユーザ要求に関連する必要な動作を実行するためにセキュリティ検査が必要であるかどうかを判定する（ステップ５２０）。ユーザが、保護資源へのアクセスを要求した場合、適当なセキュリティ検査機構を呼出し、必要な認証を達成する（ステップ５２２）。ユーザの要求には、さまざまな段階でユーザから入力データを集め、転送することが伴う場合がある。そうである場合には、ＨＴＭＬ置換変数を含むＨＴＭＬテンプレートを、全体処理またはトランザクションの一部として評価することができる（ステップ５３０）。必要なＨＴＭＬテンプレート及びＨＴＭＬ変数の処理のすべてが、必要に応じて実行される（ステップ５３２）。ある時点で、ユーザがウェブ・ブラウザの要求に応答せず、ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩが発行されて、ユーザの要求の処理が一時的に中止される場合がある。その代わりに、前にユーザ対話がすでに発生しており、ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩが以前に呼び出されている可能性もある。どちらの場合でも、現行プロセスを中止するか、前に中止されたプロセスを再開する必要がある（ステップ５４０）。必要であれば、必要に応じて適当なプロセスを再始動または中止する（ステップ５４２）。最後に、実際にユーザ要求を評価でき、ソフトウェア・アプリケーションによって適当な応答を生成することができる（ステップ５５０）。要求された機能を実行した後に、ユーザの要求によって生成された結果が、ウェブ・ブラウザによって表示される（ステップ５６０）。必要であれば、追加のウェブ・ブラウザ要求を、上に掲げたステップを繰り返すことによって同様の形で処理することができる。

【００４０】図５に示された諸ステップを、独立であり、順次であるものとして説明し、図示したが、これらのさまざまなステップは、順次である必要はなく、下のレンタ・カーの例に示されるように、同一のウェブ・トランザクション内で統合化されることが好ましい。これらのステップは、必要に応じて、任意の組み合わせまたは所望の順序で実行できる。一部のソフトウェア・アプリケーションでは、図示のステップのすべてを完了することが必要になる場合があり、他のプロセスや要求では、完了のために図示のステップのうちの１つまたは２つだけが必要になる場合がある。ステップの実際のシーケンスと各ステップの詳細な要件は、主として特定のソフトウェア・アプリケーションの設計選択のままになる。

【００４１】セキュリティ機構－ＦｌｏｗＭａｒｋの例
ここで図２及び図６を参照すると、図５のステップ５２２は、セキュリティ機構によって実行される。このステップをソフトウェア・アプリケーションのアクセスに関連して説明するが、このステップは、データベースやネットワークなどの他のコンピュータ資源へのアクセスに使用される時にも同等に適用可能である。上で述べたように、クライアント・ワークステーション２１０のユーザは、ウェブ・サーバ・アプリケーション２２２にデータを入力することによってＷＷＷを介するソフトウェア・アプリケーションへのアクセス権をウェブ・ブラウザ２１２に要求させる動作を実行する。入力データには、適当な認証データを入力するのに使用される入力空白を含むＨＴＬＭフォームを含めることができる。認証データを入力した後に、ユーザは、このデータをウェブ・サーバ・アプリケーション２２２に発行する（ステップ６２１）。好ましい実施例によれば、認証データには、ユーザＩＤ、パスワード及びキーが含まれる。ユーザＩＤ

及びパスワードには、通常のウェブ・ベースのユーザＩＤとパスワードを含めることができる。キーには、識別機構として働くデータの組を含めることができる。たとえば、可能なキーの１つが、「flowmarkuser01975」である。

【００４２】ウェブ・サーバは、ユーザＩＤとパスワードを使用して、標準的なウェブ認証手順に従ってユーザを認証する（ステップ６２３）。ユーザがウェブ・サーバ・アプリケーション２２２によって正しく認証された時には、ユーザＩＤとキーが、特定のソフトウェア・アプリケーションへのアクセスの要求と共にゲートウェイ３３２に渡される（ステップ６２５）。ゲートウェイ３３２は、ユーザＩＤ及びキーを、ユーザ・ライブラリ６２０に格納されたユーザＩＤ及びキーと比較する（ステップ６２７）。ユーザ・ライブラリ６２０には、ゲートウェイ３３２を介してアクセスされる複数のソフトウェア・アプリケーションのためのユーザ情報が格納されることが好ましい。ユーザ・ライブラリ６２０は、さまざまな形で構成できる。たとえば、各ソフトウェア・アプリケーションが、認可されたユーザの全員に対して同一のキーを有するようにすることができる。したがって、ゲートウェイ３３２は、キーを受け取った時に、そのキーによって指定されるアプリケーションに関する、ユーザ・ライブラリ６２０内の発行されたユーザＩＤとユーザを比較する。その代わりに、各キーを独自とすることができ、ゲートウェイは、一致するキー及びユーザＩＤを求めてライブラリ全体を探索することになる。

【００４３】具体的な実施態様とは無関係に、ユーザ・ライブラリ６２０には、許可ユーザごとに、対応するソフトウェア・アプリケーションにアクセスするのに必要な認証データが含まれる。したがって、ゲートウェイ３３２は、必要なソフトウェア・アプリケーション認証データを取り出すことができる（ステップ６２９）。このソフトウェア認証データを用いて、ゲートウェイ３３２は、ソフトウェア・アプリケーション３４２への通信接続を作成することができる（ステップ６３１）。具体的に言うと、好ましい実施例では、ユーザ・ライブラリ６２０から、各ソフトウェア・アプリケーションへの非ウェブ・ベース・アクセス権の認可に通常必要となる認証データと同一の認証データを提供できる。ユーザ・ライブラリ６２０には、各ソフトウェア・アプリケーションの通常の認証データが格納されるので、ゲートウェイ３３２は、ソフトウェア・アプリケーション３４２へのネイティブ・インターフェースによって提供される通常のセキュリティ手順を使用することができる。したがって、ネイティブ・ソフトウェア・アプリケーション・セキュリティ・システムを変更せず、システム管理担当者がソフトウェア・アプリケーション３４２へのウェブ・アクセスを提供するための特殊なセキュリティ・システムを実施し、維持する必要もなしに、ソフトウェア・アプリケーション３４２への安全なウェブ接続を提供することができる。さらに、ユーザは、ネイティブ・ソフトウェア・アプリケーション・セキュリティ・システムを知る必要も、理解する必要もない。というのは、ユーザはＧＵＩと対話し、このＧＵＩがネイティブ・セキュリティ・システムと対話するからである。

【００４４】この例では、キーとユーザＩＤが、ソフトウェア・アプリケーション３４２のためにユーザ・ライブラリ６２０に格納されたユーザＩＤ及びキーに対応する。ユーザＩＤとキーは、ユーザがアクセスを許可されているソフトウェア・アプリケーションの認証情報を突き止めるのに使用される。ユーザＩＤとキーが、ユーザ・ライブラリ６２０内で見つからない場合、そのユーザは、要求した資源にゲートウェイ３３２を介してアクセスすることを許可されない。ユーザ・ライブラリ６２０に格納されるソフトウェア・アプリケーション認証データには、ワークグループ・ユーザ名、ワークグループ・パスワード、アプリケーション・データベース識別子、または、ソフトウェア・アプリケーション３４２によって使用されるネイティブ・セキュリティ・システムに従うソフトウェア・アプリケーション３４２との通信を容易にするのに必要な他の情報を含めることができる。この認証データは、絶対にＷＷＷには送信されないので、許可されないユーザによるスヌーブの危険にはさらされない。

【００４５】本発明のセキュリティ機構は、ウェブ・ブラウザからＷＷＷを介するソフトウェア・アプリケーションへの安全なアクセス処理及び安全な認証処理を提供する。この機構は、複数のソフトウェア・アプリケーションの認証データを含むユーザのライブラリを使用することによって、ユーザが複数のパスワード及びユーザＩＤを記憶することを必要とせずに、複数のソフトウェア・アプリケーションまたは他のコンピュータ資源への安全なアクセスを提供する。この機構には、ウェブ・ブラウザ、ウェブ・サーバ・アプリケーション及びアプリケーション・ゲートウェイを実行し、適当なセキュリティ検査を実行した後にソフトウェア・アプリケーションを実行する、１つまたは複数のコンピュータが含まれる。この機構を用いると、ウェブ・ブラウザのユーザが、ソフトウェア・アプリケーションにアクセスできるようになる。ユーザは、ウェブ・ブラウザを介してデータを入力し、この入力データが、ウェブ・サーバ・アプリケーションに通信される。ウェブ・サーバ・アプリケーションは、ウェブ・ブラウザを認証し、ウェブ・ブラウザを一意に識別するデータを含む適当な入力データをアプリケーション・ゲートウェイに渡す。アプリケーション・ゲートウェイは、ブラウザから受け取った認証データを使用して、ブラウザのユーザがソフトウェア・アプリケーションのアクセスを許可されるかどうかを判定する。

【００４６】好ましい実施例では、ブラウザ認証情報を

使用して、ユーザとソフトウェア・アプリケーションの適当な認証情報とを相関させる。アプリケーション・ゲートウェイは、この新しい認証情報を使用して、ソフトウェア・アプリケーションへのアクセスを得る（すなわちログ・オンする）。

【００４７】図４、図６及び図９を参照すると、ＷＷＷ上でＦｌｏｗＭａｒｋアプリケーション３４２'にアクセスするのに使用される、本発明のセキュリティ機構が図示されている。上で説明したように、ＷＷＷ上でＦｌｏｗＭａｒｋアプリケーション３４２'にアクセスする必要があるユーザは、ウェブ・サーバ・アプリケーション２２２へのデータ・ストリームを生成する何らかの動作を行う。ウェブ・サーバ・アプリケーション２２２は、ユーザのデータ・ストリームを調べて、ユーザの要求を満たすために行わなければならない動作を決定する。ユーザが要求した情報が、ＦｌｏｗＭａｒｋアプリケーション３４２'へのアクセスを必要とする場合、ウェブ・ブラウザ２１２から生成されたデータ・ストリームに、ウェブ・サーバ・アプリケーション２２２がＣＧＩ４２０に制御を渡すように指令するコマンドが埋め込まれている。この例では、許可されないユーザがＷＷＷを介してＦｌｏｗＭａｒｋアプリケーション３４２'にアクセスしないようにするために、ＣＧＩ４２０へのアクセスが保護されている。ＣＧＩ４２０へのアクセス権を得るためには、ユーザは、ウェブ・サーバ・アプリケーション２２２によって認証されなければならない。ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２によって生成されたデータ・ストリーム内にＣＧＩコマンドを見つけた時に、ウェブ・ブラウザ２１２を介してユーザからパスワードとユーザの識別（ユーザＩＤ）を要求する。

【００４８】さらに、好ましい実施例によれば、ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２にキーを要求する。ウェブ・ブラウザからウェブ・サーバへキーを提供し、配布する方法には、主に２つの方法がある。まず、キーを、ウェブ・サーバに発行される元々のＨＴＭＬページに埋め込むことができる。これは、アクセスの制御が主な問題でない場合に好ましい方法である。たとえば、ＨＴＭＬフォームにキーを埋め込むことは、発行されるＨＴＭＬフォームが、ソフトウェア・アプリケーションの機能のうちの制限された機能にアクセスするために訪問者または「ゲスト」によって使用されるものである場合に有用である。ＣＧＩに渡されるフォームにキーを埋め込むことによって、ゲートウェイは、この「ゲスト」キーに対応するソフトウェア・アプリケーションへのアクセスをユーザに許可するように自動的に指令される。さらに、この状況では、ユーザＩＤとパスワードも埋め込むことができる。ユーザＩＤ、パスワード及びキーのすべてがＨＴＭＬフォームに埋め込まれていると、ユーザがアクセスを認証されたことを意識することすらなしに、必要な認証が実現される。

【００４９】キーは、ゲートウェイがＨＴＭＬフォーム内のキーとして認識する値を変数に割り当てることによって埋め込まれることが好ましい。ウェブ・サーバ・アプリケーション２２２は、フォームを受信し、制御をＣＧＩ４２０に渡し、ＣＧＩ４２０は、変数とそれに割り当てられた値を解析し、この情報をＦｌｏｗＭａｒｋ／インターネット・ゲートウェイ（ＦＭＩＧ）４３０に渡す。ＦＭＩＧ４３０は、キーの値及びユーザＩＤを、ユーザ・ライブラリ６２０に格納された値と比較する。その後、ＦＭＩＧ４３０は、キーに基づいて、選択されたソフトウェア・アプリケーションの認証データを取り出す。

【００５０】キーを供給するための第２の方法では、ウェブ・サーバ・アプリケーション２２２が、ユーザがキーを入力することの要求をウェブ・ブラウザ２１２に送信する。この方法では、ユーザが値を入力する必要があり、したがって、キーに対応するソフトウェア・アプリケーションへのアクセスに追加のセキュリティがもたらされる。通常、ウェブ・サーバ・アプリケーション２２２からウェブ・ブラウザ２１２にＲＥＡＬＭ要求が発行される。ＲＥＡＬＭ要求は、ＣＧＩ４２０へのアクセス権を得るためにユーザがユーザＩＤとパスワードを入力することを要求する、周知のソフトウェア・セキュリティ機能である。

【００５１】ウェブ・ブラウザ２１２がこの要求を受信した時、ユーザは、先に進むために、ユーザＩＤ、パスワード及びキーを入力し、これらをウェブ・サーバ・アプリケーション２２２に発行しなければならない。したがって、ウェブ・ブラウザ２１２は、ユーザＩＤ、パスワード及びキーを記憶し、ウェブ・サーバ・アプリケーション２２２に送信する。その後、ウェブ・サーバ・アプリケーション２２２は、ユーザＩＤとパスワードを使用して、ウェブ・サーバ・アプリケーション２２２へのアクセス権を有するユーザとしてそのユーザを認証する。ウェブ・サーバ・アプリケーション２２２は、ユーザから受信したユーザＩＤ及びパスワードを記憶し、このユーザＩＤとキーを、ＣＧＩ４２０を介してＦＭＩＧ４３０に送る。ＦＭＩＧ４３０は、ユーザＩＤ及びキーを、ユーザ・ライブラリ６２０の値と比較する。一致が見つかる場合、ＦＭＩＧ４３０は、そのユーザＩＤ及びキーに対応する必要な認証データを、ユーザ・ライブラリから取り出す。

【００５２】好ましい実施例では、認証データに、ＦｌｏｗＭａｒｋの非ウェブ・ベースのネイティブ・インターフェースを介してＦｌｏｗＭａｒｋにアクセスするのに通常必要となる情報のすべてが含まれる。したがって、ユーザ・ライブラリに格納される認証データには、ＦｌｏｗＭａｒｋワークグループ・ユーザ名、Ｆｌｏｗ

Ｍａｒｋワークグループ・パスワード、ＦｌｏｗＭａｒｋアプリケーション・データベース識別子、または、発行されたキーに対応するＦｌｏｗＭａｒｋアプリケーションとの通信を簡単にするのに必要な他の情報を含めることができる。この認証データは、絶対にWWWには送信されないので、スヌーブの危険にはさらされない。

【００５３】ＦｌｏｗＭａｒｋと共に使用するのに適した書式を有する、ユーザ・ライブラリ６２０の一部９００を図９に示す。ユーザ（たとえばＵｓｅｒ　ａ、Ｕｓｅｒｂ）ごとに、リストされたユーザのそれぞれがＦｌｏｗＭａｒｋアプリケーション３４２'にアクセスするのに必要な認証情報を提供する、対応するユーザ名、パスワード、データベース・サーバ識別子及びデータベース識別子がある。ユーザ・ライブラリ６２０には、ＦｌｏｗＭａｒｋ用の通常の認証データが格納されるので、ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋに接続する時に通常のセキュリティ手順を使用することができる。したがって、ＦｌｏｗＭａｒｋへの安全なウェブ接続を、通常のＦｌｏｗＭａｒｋセキュリティ・システムを変更する必要なしに実現できる。システム管理担当者は、ＦｌｏｗＭａｒｋへのウェブ・アクセスのために特殊なセキュリティ・システムを実施し、維持する必要がない。本発明のセキュリティ機構を用いると、ＦＭＩＧ４３０が、ユーザを認識できるようになり、ＦｌｏｗＭａｒｋへの接続を確立できるようになる。

【００５４】インターフェース構成要素機構

図２及び図７を参照すると、図５のステップ５３２は、ＨＴＭＬ変数とＨＴＭＬテンプレートを使用するインターフェース構成要素機構によって実行される。上で述べたように、ユーザは、ウェブ・サーバ・アプリケーション２２２にデータを入力することによってWWWを介するソフトウェアへのアクセス権をウェブ・ブラウザに要求させる動作を実行する（ステップ７１８）。この例では、入力データに、ＨＴＭＬテンプレートの位置を指定するＵＲＬまたは他のアドレス・データが含まれる。ウェブ・サーバ・アプリケーション２２２は、指定されたテンプレートのＵＲＬアドレスをゲートウェイ３３２に渡す（ステップ７２０）。ゲートウェイ３３２は、テンプレート・ライブラリ７１９内で突き止められた、指定されたＨＴＭＬテンプレートを取り出し、オーブンする（ステップ７２１）。テンプレート・ライブラリ７１９は、ゲートウェイ３３２と同一の位置に配置されることが好ましいが、ゲートウェイ３３２によってアクセス可能である限り、どこにでも配置できる。テンプレート・ライブラリ７１９には、システム開発者がソフトウェア・アプリケーション３４２のインターフェースを開発する際に例として使用できる、広範囲のＨＴＭＬテンプレートが含まれることが好ましい。

【００５５】図示の例では、ＵＲＬアドレスによって、テンプレート・ライブラリ７１９のテンプレート２が指定される。その後、ゲートウェイ３３２が、テンプレート２に格納された変数を識別する（ステップ７２３）。ゲートウェイ３３２は、ソフトウェア・アプリケーション３４２が理解できる適当なコマンドを生成し、そのコマンドをソフトウェア・アプリケーション３４２に通信する。その後、ゲートウェイ３３２は、置換変数のそれぞれに必要なデータを、ソフトウェア・アプリケーション３４２に要求する（ステップ７２５）。その後、ソフトウェア・アプリケーション３４２は、入力データの要求を処理する（ステップ７２７）。入力データの要求を処理する際に、ソフトウェア・アプリケーション３４２は、データの生成、データを有する他のプログラムの呼出し、または、ローカルまたはネットワーク上の記憶装置からのデータ検索のために、追加のソフトウェア・プロセスを開始することができる。その後、ソフトウェア・アプリケーション３４２は、要求されたデータをゲートウェイ３３２に返す（ステップ７２７）。ゲートウェイ３３２は、ＨＴＭＬテンプレート内の変数を、ソフトウェア・アプリケーション３４２から取り出したデータに置換する（ステップ７２９）。ゲートウェイ３３２は、置換変数を実際のデータに置換されたＨＴＭＬテンプレートをウェブ・サーバ・アプリケーション２２２に出力する（ステップ７３１）。ウェブ・サーバ・アプリケーション２２２は、ウェブ・サーバ出力データをウェブ・ブラウザ２１２に供給する（ステップ７３３）。

【００５６】ソフトウェア・アプリケーション３４２とウェブ・サーバ・アプリケーション２２２の間のゲートウェイとしてゲートウェイ３３２を使いやすくするために、ＨＴＭＬテンプレートのライブラリを設ける。ＨＴＭＬテンプレートのライブラリによって、複数のＣＧＩモジュールを必要とせずに、ソフトウェア・アプリケーション３４２を含む複数のアプリケーションへのウェブ・ブラウザ２１２を介するアクセスを提供する、柔軟で簡単にカストマイズできる方法がもたらされる。したがって、システム操作員は、ＣＧＩプログラミング要件を満足する必要なしに、適当なテンプレートを使用することによって、広範囲のアプリケーションに対するカスタム・ウェブ・インターフェースを作成することができる。

【００５７】好ましい実施例では、供給されるＨＴＭＬテンプレートのそれぞれに、１つまたは複数の変数が含まれる。テンプレートを指定する実行依頼をウェブ・クライアントから受信した時に、ゲートウェイ３３２がテンプレートをオーブンする。ゲートウェイ３３２は、テンプレートを解析し、テンプレート内の変数を突き止める。その後、これらの変数の値を、それに関連するソフトウェア・アプリケーションに要求する。ソフトウェア・アプリケーション３４２は、単純に変数に対応するデータを検索することができ、また、適当なデータを生成するためにソフトウェア・プロセスを開始することもで

きる。複数のソフトウェア・アプリケーションに関連する変数を挿入することによって、単一のウェブ・ページによって、これらの複数のソフトウェア・アプリケーションからのデータにアクセスすることができる。データは、ゲートウェイ３３２によってＨＴＭＬテンプレートに解析され、ウェブ・サーバ・アプリケーション２２２に配布され、ウェブ・サーバ・アプリケーション２２２は、この出力をウェブ・ブラウザ２１２に配布する。したがって、正しく定義された変数を有するテンプレートを使用することによって、システム操作員は、すべてのソフトウェア・アプリケーションのために完全に新規のカスタム・インターフェースを作成するという苦労なしに、ソフトウェア・アプリケーション・データへのウェブ・ベースのアクセスを提供することができる。

【００５８】好ましい実施例では、ＨＴＭＬテンプレートのライブラリに、ソフトウェア・アプリケーションとの対話をウェブ・ブラウザに提供するためにシステム開発者が使用できる広範囲のＨＴＭＬページが含まれる。この目的を達成するために、ＨＴＭＬテンプレートに、使用されるＨＴＭＬの複雑さのレベルが異なるさまざまなテンプレートが含まれることが好ましい。たとえば、一部のテンプレートでは、ＨＴＭＬバージョン３．２でのみ指定されているタグなどの高度なＨＴＭＬタグを使用することができる。これらの高度なタグを使用することによって、洗練された書式設定が可能になるが、それに対応した高度なウェブ・ブラウザが必要になる。したがって、これらの高度なタグを含むＨＴＭＬテンプレートは、ユーザがソフトウェア・アプリケーションのアクセスに高度なウェブ・ブラウザを使用すると期待するソフトウェア開発者によって使用される可能性がある。逆に、ＨＴＭＬ　２．０などのＨＴＭＬの以前のバージョンに忠実であり、したがって、広範囲の高度でないウェブ・ブラウザによってアクセスできるＨＴＭＬテンプレートも提供できる。広範囲のテンプレートを提供することによって、好ましい実施例では、システム操作員がソフトウェア・アプリケーションへのウェブ・インターフェースを提供する際の高い柔軟性がもたらされる。

【００５９】好ましい実施例のＨＴＭＬテンプレートには、ウェブ・ブラウザ２１２とソフトウェア・アプリケーション３４２の間でデータを渡すのに使用される入力変数が含まれる。さらに、ソフトウェア・アプリケーション３４２からのカストマイズされた出力を提供するために、置換変数が使用される。これらの変数は、ゲートウェイ３３２によって識別できる形でＨＴＭＬテンプレートに書き込まれる。このような方法の１つでは、変数をＨＴＭＬコメントに書き込み、コメントに配置された変数を解析できるウェブ・サーバを使用する。

【００６０】入力変数は、ウェブ・ブラウザ２１２からゲートウェイ３３２及びソフトウェア・アプリケーション３４２へ入力を提供するためにＨＴＭＬページに挿入される。たとえば、次にウェブ・ブラウザ２１２に送信するＨＴＭＬページを指定する入力変数を、ＨＴＭＬページに含めることができる。この場合、そのＨＴＭＬページがウェブ・サーバ・アプリケーション２２２に送り返された時に、ゲートウェイ３３２が、変数を解析し、指定されたＨＴＭＬページをウェブ・ブラウザ２１２に配布するようにウェブ・サーバ・アプリケーション２２２に指示する。

【００６１】さらに、ゲートウェイ３３２は、あるＨＴＭＬ画面から次のＨＴＭＬ画面へ特定の変数を渡すように構成することができる。たとえば、特定の通信インスタンスを表す変数を後続のＨＴＭＬ画面に挿入して、ゲートウェイ３３２が、特定のユーザのソフトウェア・アプリケーション３４２との対話を追跡できるようにすることができる。

【００６２】置換変数とは、動的データをウェブ・ブラウザに送り返すのに使用される、ＨＴＭＬテンプレートに含まれる変数である。これらの変数は、ゲートウェイ３３２によってＨＴＭＬテンプレートから解析される。その後、置換変数は、ソフトウェア・アプリケーション３４２に渡される。ソフトウェア・アプリケーション３４２は、置換変数によって表される適当なデータを突き止める。このデータは、定義済みのプロセスに従ってソフトウェア・アプリケーション３４２によって生成することができ、また、単純に適当なデータ記憶装置から取り出すことができる。

【００６３】システム操作員は、ユーザ定義変数をＨＴＭＬテンプレートに追加して、異なるタイプのソフトウェア・アプリケーションのために必要な特化された入力及び出力を提供することができる。これらの変数は、変数情報が渡されるソフトウェア・アプリケーションまたは変数情報が取り出されるソフトウェア・アプリケーションをゲートウェイ３３２が決定できる形で命名されなければならない。さらに、ＨＴＭＬテンプレートに複数の変数を追加することによって、複数の異なるソフトウェア・アプリケーションへの入力を、ウェブ・ブラウザ２１２からの１つのテンプレートの発行によって行うことができる。同様に、データを複数のソフトウェア・アプリケーション３４２から取り出し、単一のウェブ・ページに挿入してウェブ・ブラウザ２１２に出力することができる。

【００６４】切断機構

図３及び図８を参照すると、図５のステップ５４２は、切断機構によって実行される。処理のこの位置では、クライアント・ワークステーション２１０とソフトウェア・アプリケーション３４２の間のゲートウェイ３３２を介する通信がすでに確立されている。これには、一般に、必要であれば、上で説明したウェブ・サーバの認証手段及びセキュリティ手段を使用するユーザの認証が含まれる。さらに、ユーザとソフトウェア・アプリケーシ

ョン３４２の間でこの「会話」または特定の通信を識別するために、識別子が作成されている。この識別子は、会話識別子と称し、ユーザとソフトウェア・アプリケーション３４２の間で伝送されるすべての情報に付加され、この特定の通信に属するものとしてその情報にタグを付けるのに使用される。この会話識別子によって、特定のウェブ・ブラウザとソフトウェア・アプリケーション３４２の間のすべての会話に一意のマークまたはタグが付けられる。会話は、一般に、ＯＰＥＮアプリケーション・プログラム・インターフェース（ＡＰＩ）の呼出しによって開始され、一般に、ＣＬＯＳＥＡＰＩの呼出しによって終了する。ゲートウェイ３３２は、会話識別子を使用して個々の要求を記憶する。というのは、ゲートウェイ３３２が、複数のユーザと、これらのユーザのそれぞれからの複数の要求をサービスする可能性があるからである。したがって、ゲートウェイ３３２は、さまざまなユーザとソフトウェア・アプリケーション３４２の間のデータの流れを制御するのに必要な情報を維持する。

【００６５】一般に、あるユーザがソフトウェア・アプリケーション３４２にアクセスした後に、ソフトウェア・アプリケーション３４２は、そのユーザの要求が完了するか、ウェブ・クライアントからの次の入力が必要になるまで、そのユーザの要求の処理を続ける（ステップ８１９）。たとえば、次の処理に必要な追加データを突き止めるために適当なデータベースを選択するため、ソフトウェア・アプリケーション３４２がユーザからの入力を必要とする場合に、処理が停止される可能性がある。この時点で、ソフトウェア・アプリケーション３４２は、ゲートウェイ３３２を介してユーザに入力の要求を送信する（ステップ８２１）。その後、ソフトウェア・アプリケーション３４２は、ゲートウェイ３３２に対するＤＩＳＣＯＮＮＥＣＴ　ＡＰＩを呼び出す（ステップ８２１）。ＤＩＳＣＯＮＮＥＣＴＡＰＩは、ソフトウェア・アプリケーション３４２が入力を待っている間にプロセスを中止できるようにするのに使用される切断機構である。ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩは、ゲートウェイ３３２に、会話識別子を含む、中止される会話に対応する必要なデータ及び状態情報を保管させる。その後、ソフトウェア・アプリケーション３４２は、現在のソフトウェア・プロセスを中止し、後程ソフトウェア・プロセスを再始動できる状態に戻る。

【００６６】その後、ゲートウェイ３３２は、ウェブ・サーバ・アプリケーションを介して入力の要求をユーザに通信する（ステップ８２３）。ユーザは、通常、適当なデータをＨＴＭＬフィールドに入力することによって、この要求を満たす。ウェブ・クライアントのユーザが、発行ボタンを押した時に、入力されたデータが、ウェブ・サーバとゲートウェイ３３２を介してソフトウェア・アプリケーション３４２に送り返される。この応答は、すぐに返される場合も、比較的長い時間が経過した後に返される場合もある。どの場合でも、ゲートウェイ３３２は、入力データを送信したウェブ・クライアントを識別し、そのデータが中止されているソフトウェア・アプリケーション３４２のプロセスのためのものであることを認識する。これは、入力に含まれる会話識別子を、その会話識別子によってインデクシングされる会話のデータベースと突き合わせることによって行われることが好ましい。その後、ゲートウェイ３３２は、会話識別子に対応する中止されたプロセスを再始動するようにソフトウェア・アプリケーションに指示する（ステップ８３１）。その後、ソフトウェア・アプリケーション３４２は、中止されたプロセスを中止された位置から再始動し、入力データを受け取る（ステップ８３３）。この時点で、プロセスは、必要に応じて、前のように追加処理のためにステップ８１９に戻るか、終了する。

【００６７】好ましい実施例では、ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩを呼び出し、ソフトウェア・アプリケーション３４２のプロセスを中止するステップが、ウェブ・ブラウザ２１２からの入力が必要になるたびに行われる。これによって、入力が返されるのを待つプロセスのために必要なコンピュータ資源の量が制限される。ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩは、保守のためにウェブ・サーバ・コンピュータ・システム２２０、コンピュータ・システム３３０またはコンピュータ・システム３４０を一時的に遮断する必要がある時に必ず使用されることが好ましい。これは、走行中のソフトウェア・プロセスを不当に破壊せずに信頼性を確保するために使用することができる。

【００６８】ゲートウェイ３３２には、ウェブ・ブラウザ２１２とソフトウェア・アプリケーション３４２の間の対話を簡単にするために設計された複数のＡＰＩが含まれる。このＡＰＩには、ＯＰＥＮ　ＡＰＩとＤＩＳＣＯＮＮＥＣＴ　ＡＰＩが含まれることが好ましい。さらに、このＡＰＩに、ＳＥＮＤ　ＡＰＩ、ＲＥＣＥＩＶＥ　ＡＰＩ及びＣＬＯＳＥ　ＡＰＩを含めることができる。

【００６９】ＯＰＥＮ　ＡＰＩを呼び出すと、ソフトウェア・アプリケーション３４２とウェブ・ブラウザ２１２の間の１組の対話が開始される。具体的に言うと、ＯＰＥＮ　ＡＰＩは、呼び出された時に、ソフトウェア・アプリケーション３４２とウェブ・ブラウザ２１２の間の会話にタグを付けるのに使用される会話識別子を生成する。会話には、所望のタスクを達成するために複数の異なるプロセスを含めることができる。会話識別子は、個々の通信のそれぞれがどの会話に属するかを識別するために、すべての通信（すなわち、ソフトウェア・アプリケーション３４２とゲートウェイ３３２の間の通信と、ウェブ・ブラウザ２１２とウェブ・サーバ・アプリケーション２２２の間の通信）に含まれることが好まし

い。

【００７０】ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩを用いると、ソフトウェア・アプリケーション３４２が、必要になるか再始動が可能になるまで、走行中のプロセスを中止できるようになる。これは、ソフトウェア・アプリケーション３４２のすでに走行しているプロセスを完了するためにウェブ・クライアント・ユーザからの入力が必要な場合に特に有用である。ソフトウェア・アプリケーション３４２がこのような入力を必要とする時には、ウェブ・サーバ・アプリケーション２２２を介してウェブ・ブラウザ２１２に要求を送り返さなければならない。さまざまな理由から、入力が送信されるとしても、その入力がウェブ・サーバ・アプリケーション２２２を介してソフトウェア・アプリケーション３４２に送り返されるまでに、数時間もしくは数日を要する可能性がある。ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩを用いると、ソフトウェア・アプリケーション３４２が、入力を受信するまでそのプロセスを中止できるようになる。具体的に言うと、ソフトウェア・アプリケーションは、そのプロセス（会話識別子によって識別される）を中止することの通知をゲートウェイ３３２に送り、ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩを呼び出す。ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩは、中止されるプロセスの再始動に必要な情報を保存するようにゲートウェイ３３２に指示する。したがって、最終的に入力が受信された時に、ゲートウェイ３３２は、この入力が中止されたプロセス用であることを認識し（関連する会話識別子を使用することによって）、中止されたプロセスを再始動するようにソフトウェア・アプリケーション３４２に指示し、その入力と必要な状態情報をソフトウェア・アプリケーション３４２に送り返す。

【００７１】したがって、ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩを用いると、ソフトウェア・アプリケーション３４２とウェブ・ブラウザ２１２の間の通信の長い遅延が可能になる。ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩを用いなければ、この長い遅延によって、動作を待っている状態のまま未完了のソフトウェア・プロセスがもたらされる可能性がある。ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩは、会話識別子を受け取り、必要な時にゲートウェイ３３２がプロセスを再起動できるように、必要な情報を保管する。

【００７２】ＣＬＯＳＥ　ＡＰＩは、特定の会話について、ソフトウェア・アプリケーション３４２とウェブ・ブラウザ２１２の間の対話のすべてが完了した時に発行される。ソフトウェア・アプリケーション３４２が、ＣＬＯＳＥ　ＡＰＩを呼び出し、これによって、その会話に完了のタグが付けられ、クローズされた会話に関係する不要なファイル及びデータが除去される。

【００７３】ＳＥＮＤ　ＡＰＩは、ウェブ・ブラウザ２１２にデータを（ＨＴＭＬページまたは他のタイプの形式で）送信するために発行される。ソフトウェア・アプリケーション３４２が、ＳＥＮＤ　ＡＰＩを呼び出し、これによって、ゲートウェイ３３２とウェブ・サーバ・アプリケーション２２２を介してウェブ・ブラウザ２１２に送信要求が送信され、データが送信される。ＳＥＮＤ　ＡＰＩでは、会話識別子を使用して、要求されたデータを必要とするウェブ・ブラウザ２１２を判定する。

【００７４】ＲＥＣＥＩＶＥ　ＡＰＩは、ウェブ・ブラウザ２１２からデータを受信するために発行される。これには、必要に応じてＳＴＤＩＮデータと環境データを含めることができる。ソフトウェア・アプリケーション３４２は、ＲＥＣＥＩＶＥＡＰＩを呼び出して、入力を待つようにゲートウェイ３３２に指示する。入力が到着した後に、ゲートウェイ３３２は、会話識別子を使用して入力データを適当な会話と突合せ、必要な処理のためにソフトウェア・アプリケーション３４２にそのデータを渡す。したがって、ゲートウェイ３３２は、ソフトウェア・アプリケーション３４２とウェブ・ブラウザ２１２の間の通信を容易にするために、ＯＰＥＮ　ＡＰＩ、ＤＩＳＣＯＮＮＥＣＴ　ＡＰＩ、ＣＬＯＳＥ　ＡＰＩ、ＳＥＮＤ　ＡＰＩ及びＲＥＣＥＩＶＥ　ＡＰＩを使用する。

【００７５】ＷＷＷトランザクション・サポート機構

図４を参照すると、ＷＷＷトランザクション・サポートを説明するために、ウェブ・ブラウザ２１２からＷＷＷを介して特定のソフトウェア・アプリケーションにアクセスできるようにするシステム４００が示されている。この例では、ソフトウェア・アプリケーション３４２が、特定の機能を実行するようにプログラムされた、ＦｌｏｗＭａｒｋと称するワークフロー・アプリケーションである。この特定の例では、ＦｌｏｗＭａｒｋを使用して本発明を説明するが、本発明は、ＦｌｏｗＭａｒｋに制限されるものではない。他のソフトウェア・アプリケーションを本発明と共に使用することができ、追加のアプリケーション・ソフトウェア・パッケージがそのように使用されることが予期されている。

【００７６】ＦｌｏｗＭａｒｋは、比較的複雑なプロジェクトまたはタスクを、一連のより小さいプロセスまたはタスクに分解できるようにする、人気のあるプロセス・エンジニアリング・ツールである。ＦｌｏｗＭａｒｋによって実行されるアクティビティは、プロセスのフローとアクティビティを記述する１つまたは複数のプロセス・モデル４４０を使用することによって記述される。これによって、ＦｌｏｗＭａｒｋは、通信ネットワークを介してコンピュータ・システムによって達成できる動作及びトランザクションに非常に有用になる。ＦｌｏｗＭａｒｋは、一連のコンピュータ化されたシーケンスとして現実のタスクを記述するプロセス・モデル４４０の作成に使用される。情報は、ＦｌｏｗＭａｒｋワークフロー・ソフトウェアによって処理され、通常は、複数の関連アクティビティが伴う。ＦｌｏｗＭａｒｋワークフロー・プロセス言語によって、アクティビティがモデル

化され、適当な場合にはトランザクションが自動化される。個々のタスクは、「ワーク・リスト」（すなわち、電子「作業予定」リスト）内に生成される。ＦｌｏｗＭａｒｋを使用して特定のタスクを達成する例を、下で詳細に示す。

【００７７】図４のシステム４００は、図３に示されたコンピュータ・システム３００の１つの適当な詳細な実施態様を表す。同一のコンピュータ・システムが存在し（すなわち、クライアント・ワークステーション２１０、ウェブ・サーバ・コンピュータ・システム２２０、コンピュータ・システム３３０及びコンピュータ・システム３４０）、ウェブ・ブラウザ２１２がクライアント・ワークステーション２１０に常駐し、ウェブ・サーバ・アプリケーション２２２がウェブ・サーバ・コンピュータ・システム２２０に常駐する。さらに、図３の接続３２６には、図４の接続４２６及び４２８が含まれる。図４のシステム４００は、ＦｌｏｗＭａｒｋアプリケーション３４２'との通信に使用される時のゲートウェイ３３２の具体的な実施例を示すために本明細書に提示されたものである。

【００７８】この特定の実施例について、ゲートウェイ３３２には、ＣＧＩ４２０、ＦＭＩＧ４３０及びWWWアプリケーション・プログラム・インターフェース（ＡＰＩ）４３４が含まれる。４つの標準的なWWW　ＡＰＩが、Ｏｐｅｎ、Ｃｌｏｓｅ、Ｓｅｎｄ及びＲｅｃｅｉｖｅである。WWW　ＡＰＩ４３４によって、ＦＭＩＧ４３０とアクティビティ・プログラム４３２の間の通信リンクがもたらされる。ＦｌｏｗＭａｒｋアプリケーション３４２'には、ＦｌｏｗＭａｒｋ４５０と、１つまたは複数のアクティビティ・プログラム４３２が含まれる。ＦｌｏｗＭａｒｋ４５０には、ＦｌｏｗＭａｒｋデータベース４３８、１つまたは複数のプロセス・モデル４４０及びＦｌｏｗＭａｒｋ　ＡＰＩ４３６が含まれる。特定のソフトウェア・アプリケーション３４２は、ＦｌｏｗＭａｒｋ内でプロセス・モデル４４０を定義し、プロセス・モデル４４０内の特定のタスクを実行するアクティビティ・プログラム４３２を定義することによって、システム４００内で実施される。ＦｌｏｗＭａｒｋ　ＡＰＩ４３６は、ＦｌｏｗＭａｒｋ４５０と共に供給される標準ＡＰＩであり、これによって、アクティビティ・プログラム４３２とＦＭＩＧ４３０がＦｌｏｗＭａｒｋ４５０と通信する手段がもたらされる。ＦｌｏｗＭａｒｋデータベース４３８は、プロセス・モデルに関する情報を格納するのに使用することができる、汎用データベースである。たとえば、プロセス・モデル４４０とアクティビティ・プログラム４３２によって、レンタ・カー・ワーク・フロー・プロセスが実施される場合、ＦｌｏｗＭａｒｋデータベース４３８は、どの自動車が使用可能であるかなど、レンタ・カー・プロセスに関する情報の格納に使用されるはずである。

【００７９】WWWを介してＦｌｏｗＭａｒｋアプリケーション３４２'にアクセスする必要があるユーザは、クライアント・ワークステーション２１０を使用してウェブ・ブラウザ２１２に要求を入力する。ユーザは、特定のホーム・ページ・サイトのＵＲＬを入力するか、ウェブ・ブラウザ２１２を使用する、ＨＴＭＬによって生成されたユーザ・インターフェースに提示されるボタンをクリックすることができる。ユーザが、通常はＨＴＭＬフォームに表示されたボタンのクリックによって、必要な情報を「発行」する時に、ウェブ・サーバ・アプリケーション２２２が、ウェブ・ブラウザ２１２から入力データを受信する。このデータ・ストリームは、ＲＦＣ１８６６形式やＲＦＣ１８６７形式を含む多数の異なる形式でウェブ・サーバ・アプリケーション２２２に提示できる。この２つの具体的な形式は、一般的なウェブ・ブラウザが理解する共通データ・ストリーム形式の例にすぎない。本発明は、これらの形式に制限されるものではなく、現在既知であるか将来に開発されるデータ伝送形式のすべてを含む。

【００８０】ユーザの要求した情報のために、ＦｌｏｗＭａｒｋアプリケーション３４２'へのアクセスが必要な場合、ＣＧＩ４２０へのアクセスの必要を識別するコマンドがデータ・ストリームに埋め込まれており、このＣＧＩ４２０が、ＦｌｏｗＭａｒｋアプリケーション３４２'へのアクセスを提供する。この特定の例では、このコマンドがＣＧＩ４２０の呼び出しである。ＣＧＩ４２０の呼出しは、ＵＲＬの一部とすることができ、その代わりに、ウェブ・ブラウザ２１２によって発行されるデータに埋め込むことができる。ＣＧＩ４２０へのアクセスは、許可されないユーザがWWWを介してＦｌｏｗＭａｒｋアプリケーション３４２'にアクセスしないようにするために、保護されている。ＣＧＩ４２０へのアクセス権を得るには、ユーザは、ウェブ・サーバ・アプリケーション２２２によって認承されなければならない。図６に関して上で述べた認証機構が、好ましい認証機構である。ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２によって生成されたデータ・ストリーム内でＣＧＩコマンドを見つけた時に、ウェブ・ブラウザ２１２を介してユーザにパスワードとユーザ識別（ユーザＩＤ）を要求する。ユーザが認証された後に、ウェブ・サーバ・アプリケーション２２２は、ＣＧＩ４２０に制御を渡して、必要な動作を実行する。

【００８１】ＣＧＩは、当業者に周知の実行可能プログラムである。ＣＧＩは、WWWを介して情報を送信するための機構を提供するように機能する。ＣＧＩ４２０は、ウェブ・サーバ・アプリケーション２２２からＦＭＩＧ４３０へ、コマンドの形で情報を渡す。このコマンドは、一方向すなわちＣＧＩ４２０からＦＭＩＧ４３０に向かうが、データと状況は、両方向で受け渡しされる。ウェブ・サーバ・アプリケーション２２２は、ＣＧ

Ｉ４２０を呼び出し、要求に関する適当なコマンドとデータを送る。本発明のこの好ましい実施例では、ウェブ・ブラウザ２１２からＣＧＩ４２０へのデータ伝送に「ＣＧＩ　Ｐｏｓｔ」（ｓｔｄｉｎ）形式を使用するが、ウェブ・ブラウザ２１２によって生成できる他のすべてのデータ伝送形式が予期され、本発明の範囲に含まれる。さらに、ＣＧＩ４２０の解析機能及び他の動作機能は、他の形で実施できることに留意されたい。たとえば、多数のウェブ・サーバ・アプリケーションが、現在「モジュール」をサポートしている。モジュールとは、動的リンク・ライブラリ（ＤＬＬ）を使用することによって実施されるソフトウェア・ルーチンである。モジュールは、図４のシステム４００内でＣＧＩ４２０と同一の機能を実行することができ、本発明の好ましい実施例のいくつかでは、ＣＧＩ４２０と置換できる。モジュール及びＤＬＬの使用は、当業者に周知である。したがって、ＣＧＩ４２０の使用は、例示のみを目的とし、本発明の制限ではない。

【００８２】ウェブ・ブラウザ２１２からデータを受信した後に、ＣＧＩ４２０は、データを解析して、ＦｌｏｗＭａｒｋアプリケーション３４２'へのアクセスの要求を含む、要求された処理に関する関連情報を突き止める。ＣＧＩ４２０は、ユーザのデータ及び要求を、いくつかの制御情報と共にＦＭＩＧ４３０に送る。ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋアプリケーション３４２'がＷＷＷを介してウェブ・ユーザと対話する方法を提供する。ＦＭＩＧ４３０は、ＣＧＩ４２０とＦｌｏｗＭａｒｋアプリケーション３４２'の間の情報の流れを指示し、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を使用することによってＦｌｏｗＭａｒｋ機能を開始する。たとえば、ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を呼び出して、ユーザによって発行された要求を処理するのに必要なプロセス・インスタンスを作成することができる。その後、ＦＭＩＧ４３０は、異なるＦｌｏｗＭａｒｋＡＰＩ４３６を使用して、このプロセス・インスタンスの呼出しまたは起動を行うことができる。このプロセスは、プロセス・モデル４４０によって支配され、プロセス・モデル４４０は、所望のタスクを実行するために呼び出さなければならないアクティビティ・プログラム４３２をＦｌｏｗＭａｒｋ４５０に知らせる。ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋプロセスを起動した後に、要求が処理されたことを示す情報を、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を介してＦｌｏｗＭａｒｋ４５０から受け取るか、ＷＷＷ　ＡＰＩ４３４を介してアクティビティ・プログラム４３２から受け取るまで待機する。ＦＭＩＧ４３０とＦｌｏｗＭａｒｋ　ＡＰＩ４３６の間のコマンド・インターフェースは、一方向すなわち、必ずＦＭＩＧ４３０がＦｌｏｗＭａｒｋ　ＡＰＩ４３６を呼び出すが、データと状況情報は、両方向に流れる。これによって、ウェブ・インターフェースが変更された場合であっても、ＦｌｏｗＭａｒｋアプリケーション３４２'へのインターフェースを無変更のままにすることができるので、これは重要である。

【００８３】また、ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋとの対話を必要とする各ウェブ・クライアントからの要求に、会話識別子を割り当てる。上で説明したように、会話識別子によって、特定のウェブ・ブラウザとＦｌｏｗＭａｒｋの間のすべての会話に一意のマークまたはタグが付けられる。会話は、一般にＯＰＥＮアプリケーション・プログラム・インターフェース（ＡＰＩ）を呼び出すことによって開始され、一般にＣＬＯＳＥ　ＡＰＩを呼び出すことによって終了する。ＦＭＩＧ４３０は、複数のユーザとそれらのユーザのそれぞれからの複数の要求をサービスする可能性があるので、会話識別子を使用して個々の要求を記憶する。したがって、ＦＭＩＧ４３０は、さまざまなユーザとＦｌｏｗＭａｒｋによって処理されているプロセス・インスタンスとの間の情報の流れを制御するのに必要な情報を維持する。

【００８４】アクティビティ・プログラム４３２は、ユーザが要求したタスクを達成するためにＦｌｏｗＭａｒｋが使用できるソフトウェア・モジュールである。個々のアクティビティ・プログラム４３２は、ＦｌｏｗＭａｒｋによって開始され、その後、そのアクティビティ・プログラム４３２は、ＷＷＷ　ＡＰＩ４３４を介してウェブ・クライアントと通信する。各ワークフローのプロセス・モデル４４０によって、所望のタスクを達成するのに必要なアクティビティ・プログラム４３２が開始される。各アクティビティ・プログラム４３２は、所望のタスクを達成するために走行し、要求された情報を返し、その後終了するプログラムのインスタンスである。たとえば、アクティビティ・プログラム４３２は、ユーザの要求を処理するためにユーザに情報を要求する場合がある。アクティビティ・プログラム４３２は、適当なＷＷＷ　ＡＰＩ４３４を呼び出して、必要なデータを得る。この場合、ＦＭＩＧ４３０は、アクティビティ・プログラム４３２への発行を待っているデータを有し、このデータは、この処理でＣＧＩ４２０が前に送ったデータと同一である。アクティビティ・プログラム４３２は、ＷＷＷ　ＡＰＩ４３４を呼び出して、データの要求をＦＭＩＧ４３０に送り、ＦＭＩＧ４３０は、ＷＷＷ　ＡＰＩ４３４からの要求に応答して、このデータをアクティビティ・プログラム４３２に返す。アクティビティ・プログラム４３２は、このデータを処理するために適当なステップを実行し、要求を満たす。アクティビティ・プログラム４３２は、ＷＷＷ　ＡＰＩ４３４、ＦＭＩＧ４３０、ＣＧＩ４２０、ウェブ・サーバ・アプリケーション２２２及びウェブ・ブラウザ２１２を介してユーザと通信する。また、アクティビティ・プログラム４３２は、ＷＷＷ　ＡＰＩ４３４から戻りコードを受け取って、ユーザが要求したデータを受信したことを検証す

る。確認を受信した後に、アクティビティ・プログラム４３２のこの特定のインスタンスは、その要求を完了し、終了する。１つまたは複数のアクティビティ・プログラム４３２の他のインスタンスが、まだ活動状態であり、他の要求を処理している可能性がある。

【００８５】ＷＷＷ　ＡＰＩ４３４は、ＦＭＩＧ４３０とアクティビティ・プログラム４３２の間の対話を提供するために働く。ＷＷＷ　ＡＰＩ４３４を用いると、アクティビティ・プログラム４３２が、ウェブ・クライアントとの間でデータ及び状況を送受信できるようになる。ＦｌｏｗＭａｒｋ　ＡＰＩ４３６が、無変更のままであることに留意することが重要である。これが重要であるのは、ウェブ・ユーザがＦｌｏｗＭａｒｋにアクセスできるようにするためにＦｌｏｗＭａｒｋ　ＡＰＩが変更されることがないからである。この意味では、ＦｌｏｗＭａｒｋは、ウェブ・クライアントからアクセスされていることを知らず、専用アプリケーションが要求された機能を実行している場合と同一の形で動作する。ＦｌｏｗＭａｒｋ　ＡＰＩは影響を受けないので、ＦｌｏｗＭａｒｋの動作は、ＦｌｏｗＭａｒｋアプリケーション３４２'が達成することをウェブ・クライアントが要求しているプロセスまたはタスクに無関係に、変更されない。複数の位置からの複数のユーザが、ＷＷＷを介してＦｌｏｗＭａｒｋにアクセスでき、ＦｌｏｗＭａｒｋインターフェースが変更されないことが保証される。場合によっては、システムが透過的なウェブ・クライアントを提示するが、これは、ユーザの要求がＦｌｏｗＭａｒｋアプリケーション３４２'によって処理されていることがユーザに示されないことを意味する。

【００８６】ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２と同一の位置に配置できることに留意されたい。さらに、必要ではないが、コンピュータ・システム３３０とコンピュータ・システム３４０は、好ましい実施例では同一のコンピュータ・システムである。最後に、図面に示された接続は、当技術分野で既知の、コンピュータ・システムを接続するためのあらゆるタイプの物理的手段または論理的手段とすることができる。これには、直接接続、インターネット接続、イントラネット接続、赤外線（ＩＲ）及び他の形態の無線接続が含まれるが、これらに制限されるものではない。コンピュータ・システムを接続するための多数の代替の方法及び材料を、本発明と共に使用するために簡単に適合できることが予期される。要するに、本明細書では複数の別々のコンピュータ・システム（たとえば図３及び図４）が開示されるが、本発明によるコンピュータ・システムには、あらゆる台数のコンピュータ・システムまたはコンピュータ・システムの組み合わせを含めることができる。本明細書の図面は、本発明の顕著な態様を例示するために示されたものであって、本発明を本明細書に示された特定の構成に制限するものと解釈してはならない。

【００８７】共通ユーザ・インターフェースを介する複数のソフトウェア・アプリケーションへの複数ユーザのアクセス

図２と図１０を参照すると、本発明のもう１つの長所は、複数のユーザがＷＷＷ上でウェブ・ブラウザ２１２を介して同時に複数のソフトウェア・アプリケーションにアクセスする時に最も明らかになる。各ウェブ・ユーザは、一般に、ＷＷＷへのアクセスに使用されるクライアント・ワークステーション２１０を有する。図１０の特定の構成では、クライアント・ワークステーション１（ＣＷ１）及びクライアント・ワークステーション２（ＣＷ２）が、ウェブ・サーバ１（ＷＳ１）に結合され、クライアント・ワークステーション３（ＣＷ３）が、ウェブ・サーバ２（ＷＳ２）に結合される。クライアント・ワークステーションＣＷ４及びＣＷ５は、ウェブ・サーバ３（ＷＳ３）に結合される。ウェブ・サーバＷＳ１及びＷＳ２の両方が、ゲートウェイ・コンピュータ・システム（ＧＣＳ）、この場合にはＧＣＳ１に結合される。図からわかるように、ＷＳ３は、ＧＣＳ２に結合される。ＧＣＳ１とＧＣＳ２は、どちらもがソフトウェア・アプリケーション・コンピュータ１（ＳＡＣ１）に結合される。さらに、ＧＣＳ２は、ソフトウェア・アプリケーション・コンピュータ２（ＳＡＣ２）にも結合される。ＳＡＣ１とＳＡＣ２の両方で、ウェブ・ユーザがＷＷＷを介するアクセスを必要とする可能性があるさまざまなソフトウェア・アプリケーションが走行している。

【００８８】本発明は、ウェブ・クライアントのそれぞれに共通グラフィカル・ユーザ・インターフェースを提供し、ＳＡＣ１及びＳＡＣ２で走行するアプリケーション・ソフトウェア・プログラムへのすべてのクライアント・ワークステーション（ＣＷ１ないしＣＷ５）によるアクセスのすべてを、要求された機能が要求通りに実行されることが保証される形で管理する。ＧＣＳは、ウェブ・クライアントとソフトウェア・アプリケーションの間でトラフィックの方向を指定する、アプリケーション・ゲートウェイを提供する。本発明のセキュリティ機構は、アプリケーション・ソフトウェアと共に動作し、さまざまなコンピュータ資源のそれぞれの認証データへのアクセスに使用されるユーザＩＤとキーを必要とする。会話のそれぞれに一意の識別子を割り当てることによって、ＧＣＳは、ＳＡＣ上で走行するソフトウェア・アプリケーションのそれぞれと、そのネイティブ・インターフェースを使用して対話することができ、その後、ソフトウェア・アプリケーションからの出力を識別し、その出力を、出力を要求したウェブ・クライアントと突き合わせることができる。本発明によるＨＴＭＬテンプレートとＨＴＭＬ変数を使用することによって、一部のウェブ・クライアントは、異なるコンピュータ・システム上

の複数のソフトウェア・アプリケーションからの情報に実際にアクセスすることができ、要求したデータを単一のウェブ・ページ内で受信することができる。どの時点でも、ウェブ・クライアントのうちの１つが、処理を中止され、ソフトウェア・アプリケーションへの入力の供給を停止する可能性がある。本発明のＤＩＳＣＯＮＮＥＣＴ　ＡＰＩ機能を使用することによって、コンピュータの時間が無用に浪費されなくなる。

【００８９】自動車レンタルの例

本発明の好ましい実施例の具体的な例を、ある人が、ＷＷＷを介してレンタ・カー代理人のＦｌｏｗＭａｒｋアプリケーション・ソフトウェアにアクセスすることによって自動車を借りたいと思っている状況を使用することによって、詳細に説明できる。図２６を参照すると、自動車レンタルの例全体のプロセス・モデル２０００には、プロセス・ステップ２０１０、２０２０、２０３０、２０４０及び２０５０が含まれる。プロセス・モデル２０００のプロセス・ステップ２０１０では、アクティビティ・プログラムが自動車レンタル情報を受け取り、次の予約番号を突き止め、その予約番号をファイルに保管し、要求者に予約番号を返し、ＦｌｏｗＭａｒｋ出力データ・コンテナに予約番号をセットする。プロセス・ステップ２０１０では、ＦｌｏｗＭａｒｋのＩｎｔｅｒｎｅｔ　Ｃｏｎｎｅｃｔｉｏｎ　ＷＷＷ　ＡＰＩ及びデータ・コンテナＡＰＩを使用する。プロセス・ステップ２０２０は、予約を満たすために使用可能な自動車があるかどうかを判定するアクティビティ・プログラムである。プロセス・モデル２０００の後続経路のどちらに進むかは、この判定の結果に依存する。予約を満たすために使用可能な自動車がある場合、プロセス・モデル２０００の次のステップは、プロセス・ステップ２０３０になる。しかし、予約要求を満たすために使用可能な自動車がない場合には、プロセス・モデルの次のステップは、プロセス・ステップ２０４０になる。プロセス・ステップ２０３０では、自動車レンタル要求に一致する、使用可能な自動車が識別され、予約要求を満たすためにスケジューリングされる。さらに、予約確認番号が生成され、要求者に転送される。プロセス・ステップ２０４０では、レンタ・カー代理人は、予約要求を満たすために別の位置からレンタ・カーを捜す試みを行うか否かを選択できる。別の位置にある使用可能なレンタ・カーを捜す場合、プロセス・モデル２０００の次のステップは、プロセス・ステップ２０５０になる。プロセス・ステップ２０５０では、予約要求を満たすために自動車をスケジューリングし、ある位置から別の位置へ移動するためのアクティビティ・プログラムを使用できる。プロセス・ステップ２０５０では、別の位置で使用可能な自動車のリストが提示され、その結果、レンタ・カー代理人が、移動するのに適した自動車を選択できるようになる。これらのステップのそれぞれを、以下で詳細に説明する。

【００９０】図４と図２６を参照すると、自動車を借りたい人またはユーザは、ウェブ・ブラウザ２１２が走行しているクライアント・ワークステーション２１０を使用することによってＷＷＷにアクセスする。ユーザは、ウェブ・ブラウザ２１２を使用して、レンタ・カー代理店のＵＲＬを入力し、レンタ・カー代理店のホーム・ページ・サイトを突き止める。その後、ユーザは、レンタ・カー代理店のウェブ・サイトで、自動車レンタルの要求を受け入れるように以前に設定された適当な区域またはページを突き止める。この例では、レンタ・カー代理店のウェブ・サイトにレンタル予約フォームがあると仮定する。自動車レンタル予約フォームを突き止めた後に、ユーザは、レンタ・カー代理店が要求する情報を入力する。この情報には、通常、顧客識別番号、パスワード、自動車を借りる都市、レンタ・カー借用希望日、借りる自動車の具体的な車種（すなわち、メーカー、モデル、サイズ）などの項目が含まれる。

【００９１】適当なレンタル予約フォームまたはテンプレートを生成するＨＴＭＬの例を、図１１に示す。ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２に送り返す必要がある次の出力として図１１のＨＴＭＬテンプレートを指定する入力をウェブ・ブラウザ２１２から受信すると仮定する。この指定は、ウェブ・ブラウザ２１２からのフォーム入力の使用を含む、さまざまな方法で行うことができる。このフォームでは、ユーザが、会員番号、姓、名、出発点の都市、州、開始日、日数及び自動車の好みを入力することを要求される。ユーザは、この情報を入力した後に、レンタル予約フォーム上の「発行」ボタンをクリックすることによって、この情報を発行する。この時点で、ウェブ・サーバ・アプリケーション２２２が、ユーザ要求によって生成されたデータ・ストリームをウェブ・ブラウザ２１２から受信する。適当な書式によって、すべての変数及び他の関連情報データを拾い上げ、図１２のデータ・ストリームのような外見のｐｏｓｔデータ・ストリーム形式でウェブ・サーバ・アプリケーション２２２に送る。上で説明したように、このデータ・ストリームは、多数の異なるデータ形式でウェブ・サーバ・アプリケーション２２２に提示することができ、本発明は特定のデータ形式には制限されない。この動作は、図２６のプロセス・ステップ２０１０に含まれる。

【００９２】ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２からのデータ・ストリームを検査して、ユーザの要求を満たすために行わなければならない動作を決定する。この特定の例では、ＨＴＭＬフォームの「<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">」という行のＣＧＩ４２０への呼出しから、ウェブ・サーバ・アプリケーション２２２は、exmp5cgi.exeと称する保護ＣＧＩ（ＣＧＩ４２０と仮定す

る）を呼び出す必要があることを知る。ウェブ・サーバ・アプリケーション２２２は、この保護ＣＧＩへの呼出しを見た時に、ウェブ・サーバ・アプリケーション２２２がＣＧＩ４２０へのアクセスを許可する前に認証が必要であることを知る。

【００９３】ＣＧＩ４２０は保護されているので、許可されないユーザがＷＷＷを介してＦｌｏｗＭａｒｋアプリケーション３４２'にアクセスできなくするために、ＣＧＩ４２０へのアクセスを制御しなければならない。ＣＧＩ４２０を指定したＨＴＭＬフォーム内のパラメータに応答して、ウェブ・サーバ・アプリケーション２２２は、ウェブ・ユーザを認証しなければならない。この認証では、ユーザがユーザＩＤとパスワードを入力する必要があり、ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２にＲＥＡＬＭ要求を発行する。ＲＥＡＬＭ要求は、ＣＧＩ４２０へのアクセス権を得るためにユーザがユーザＩＤとパスワードを入力することを要求する、周知のソフトウェア・セキュリティ機能である。ウェブ・ブラウザ２１２は、ユーザから受け取ったユーザＩＤとパスワードを記憶し、この情報をウェブ・サーバ・アプリケーション２２２に送信する。ウェブ・サーバ・アプリケーション２２２は、このユーザＩＤとパスワードを使用してユーザを認証してから、そのユーザにＣＧＩ４２０へのアクセスを許可する。ウェブ・ブラウザ２１２の将来の実施態様のいくつかでは、ウェブ・サーバ・アプリケーション２２２に送信されるデータ・ストリームにウェブ・ブラウザ２１２が何らかの形の認証データを挿入する機構が設けられる可能性が高いことに留意されたい。認証データを収集し、ウェブ・ブラウザ２１２からウェブ・サーバ・アプリケーション２２２に送信するための方法及び技法のすべてが、本発明の範囲に含まれる。

【００９４】認証の後に、ウェブ・サーバ・アプリケーション２２２は、ＣＧＩ４２０に制御を渡し、ＣＧＩ４２０は、ウェブ・ブラウザ２１２からデータを受信する。この特定の例では、ＣＧＩ４２０は、標準入力プロトコル「ｓｔｄｉｎ」を介して「ｐｏｓｔ」形式のデータ・ストリームを受信する。ｐｏｓｔ形式は、当技術分野で周知のデータ形式記述である。ｓｔｄｉｎは、ｐｏｓｔデータを読み取るファイル記述子である。本発明のこの好ましい実施例では、ｐｏｓｔデータ形式記述が使用されるが、ウェブ・ブラウザ２１２及びウェブ・サーバ・アプリケーション２２２によって使用できる他のすべてのデータ形式が予期され、本発明の範囲に含まれる。この例でウェブ・ブラウザ２１２からＣＧＩ４２０に送信されるレンタル予約フォームのｐｏｓｔデータを、図１２に示す。

【００９５】ＣＧＩ４２０は、ウェブ・ブラウザ２１２からｐｏｓｔデータを受信した後に、データを解析し、ワークフロー変数を捜す。この例では、ワークフロー変数は接頭辞ｗｆ－を有する。２つの隠されたワークフロー変数が、「wf-cgi-submit=2」及び「wf-cgi-html=＼exm＼smp＼exmp5srk.html」である。最初の変数「wf-cgi-submit=2」は、ＣＧＩ４２０に、ＦｌｏｗＭａｒｋプロセス・インスタンスを作成し、始動する必要があることを知らせる。２つの隠されていないワークフロー変数が、「wf-fmig-key」と「wf-api-proc-template="WWW.Reservation.Request」である。これらの変数は、作成し、始動しなければならないプロセス・テンプレートを示す。単一のＨＴＭＬテンプレートで複数の変数を使用して、複数のアプリケーションからデータを取り出すことができる。たとえば、３つの変数を含むＨＴＭＬテンプレートを指定できる。ある変数で、始動しなければならない特定のＦｌｏｗＭａｒｋプロセスを指定でき、他の２つの変数を使用して、ＦｌｏｗＭａｒｋアクティビティ・プログラムによってアクセス可能な異なるソフトウェア・アプリケーションから情報を取り出す。ＨＴＭＬテンプレートがウェブ・サーバ・アプリケーション２２２に発行される際に、ＣＧＩは、変数を解析し、ＦＭＩＧに配布する。ＦＭＩＧは、ｗｆ変数によって指定されたプロセスを始動するようソフトウェア・アプリケーションに指示し、２つのアプリケーション変数を適当なソフトウェア・アプリケーションに渡す。ソフトウェア・アプリケーションは、変数に対応するアプリケーションから適当なデータを取り出す。したがって、適当なＨＴＭＬ変数を使用することによって、ウェブ・クライアントは、単一のウェブ・ページを介して多数の異なるアプリケーションを用いて作業することができる。この対話は、ウェブ・クライアントにとって完全に透過的とすることができ、また、インターフェース・ウェブ・ページから明白になるようにすることができる。

【００９６】したがって、置換変数を有するＨＴＭＬテンプレートを使用することによって、単一の比較的単純なＣＧＩモジュールが、ＦＭＩＧとあいまって、ウェブ・サーバと複数のソフトウェア・アプリケーションの間の効果的なインターフェースを提供することができる。これによって、システム操作員は、複数のＣＧＩモジュールを使用する必要も、極度に複雑なＣＧＩモジュールを使用する必要もなしに、ＷＷＷを介する複数のソフトウェア・アプリケーションへの簡単にカストマイズできるウェブ・アクセスを提供できるようになる。ＣＧＩ４２０は、他の制御情報と共に、ｐｏｓｔデータと環境データをＦＭＩＧ４３０に送る。環境データは、当業者に周知の標準プログラミング技法を介して入手可能であり、標準「Ｃ」プログラミング言語パラメータとしてＣＧＩ４２０に渡すことができる。

【００９７】ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を使用してＣＧＩ４２０からの要求を通信することによって、ＣＧＩ４２０とＦｌｏｗＭａｒｋアプリケーション３４２'の間の情報の流れを指示する。こ

れらのＦｌｏｗＭａｒｋ　ＡＰＩは、ＦｌｏｗＭａｒｋ製品と共に出荷される標準ＡＰＩであり、本発明に適合するために変更する必要はない。これによって、新しいウェブ・インターフェースを設ける場合であっても、ＦｌｏｗＭａｒｋアプリケーション３４２'のネイティブ・コマンド・インターフェースを無変更のままにすることができるので、これは重要である。ＦＭＩＧ４３０とＦｌｏｗＭａｒｋアプリケーション３４２'の間のコマンド・インターフェースは一方向すなわち、必ずＦＭＩＧ４３０がＦｌｏｗＭａｒｋ　ＡＰＩ４３６を呼び出すが、データと状況情報は両方向に流れることに留意されたい。

【００９８】ＣＧＩ４２０からデータを受け取った後に、ＦＭＩＧ４３０は、データを解析し、wf-fmig-key変数からユーザＩＤ及びキーを突き止めて、対応するＦｌｏｗＭａｒｋユーザＩＤと、そのユーザにＦｌｏｗＭａｒｋアプリケーション３４２'へのアクセスを提供する他のＦｌｏｗＭａｒｋアクセス情報があるかどうかを判定する。ユーザＩＤとキーに対応するＦｌｏｗＭａｒｋアクセス情報があると仮定すると、ＦＭＩＧ４３０は、このアクセス情報を使用してＦｌｏｗＭａｒｋアプリケーション３４２'にログ・インする。ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６のうちの１つを呼び出して、ユーザによって発行された要求を処理するのに必要なプロセス・インスタンスを作成する。この例では、要求によって、「www_Reservation_Request」と称するプロセス・インスタンスが作成される。このプロセス・インスタンスは、ＷＷＷ自動車レンタル・トランザクションを処理するように特別に設計されている。その後、別のＦｌｏｗＭａｒｋＡＰＩ４３６を使用して、ＦＭＩＧ４３０が、このプロセス・インスタンスを呼び出すか始動する。また、ＦＭＩＧ４３０は、「wf-cgi-html」変数用の適当なＨＴＭＬ変数情報を保管する。「wf-cgi-html」は、ワークフロー・シーケンスが終了した後に変更すべきＨＴＭＬをＦＭＩＧ４３０に知らせる変数である。ＦｌｏｗＭａｒｋプロセスwww_Reservation_Requestの接頭辞によって、このプロセスがウェブ使用可能すなわち、アクティビティ・プログラム４３２がＷＷＷＡＰＩ４３４を使用してＦＭＩＧ４３０と通信することが、ＦＭＩＧ４３０に知らされる。ＦＭＩＧ４３０が、プロセス・インスタンスとそれを要求したウェブ・ブラウザを対応させることができることを確実にするために、ＦＭＩＧ４３０は、このウェブ・クライアントの「ハンドル」を生成し、記憶する。プロセスに応じて、このハンドルは、プロセス・インスタンス名、アクティビティ・インスタンス名及びウェブ・クライアントのセキュリティ・データのなんらかの組み合わせになる。このハンドルは、ウェブ・クライアントの一意の識別子である。その後、ＦＭＩＧ４３０は、ハンドル、プロセス・インスタンス、ユーザＩＤなどを、その内部データ・キャッシュに入力する。その後、ＦＭＩＧ４３０は、ＷＷＷ　ＡＰＩ４３４の接続を待つ。この時点で、ＣＧＩ４２０は、まだ接続３１６を介してＦＭＩＧ４３０に接続されており、ユーザの自動車レンタル要求を満たすために、要求された状況またはデータをＦＭＩＧ４３０から受け取るのを待っている。

【００９９】自動車レンタル予約プロセス・モデルが初めて作成された時に、第１のアクティビティ・プログラム４３２が、識別され、ＦｌｏｗＭａｒｋ自動車レンタル予約プロセス・モデルが呼び出された時に必ず自動的に実行されるように指定された。これは、ＦＭＩＧ４３０が自動車レンタル予約プロセス・モデルのＦｌｏｗＭａｒｋインスタンスを作成し、起動する時に、必ず、第１の自動車レンタル予約ＦｌｏｗＭａｒｋアクティビティ・プログラムであるアクティビティ・プログラム４３２が、ＦｌｏｗＭａｒｋアプリケーション３４２'によって自動的に始動されることを意味する。この特定の例では、自動車レンタル要求を処理するために一緒に働く複数の関連するアクティビティ・プログラム４３２がある。ＦｌｏｗＭａｒｋワークフロー・プロセス言語によって、自動車レンタル要求及び予約プロセスがモデル化され、適当な場合にトランザクションが自動化される。自動車レンタル要求を処理するために、ＦｌｏｗＭａｒｋアプリケーション３４２'によって、ＦｌｏｗＭａｒｋデータベース４３８内に自動車レンタル予約プロセス・モデルのインスタンスが作成され、プログラム実行クライアント（ＰＥＣ）が活動状態であるかどうかが検査される。ＰＥＣは、ＦｌｏｗＭａｒｋアプリケーション３４２'内のアクティビティの動作と流れを調整する、監視プログラム・モジュールである。ＰＥＣが活動状態になった後に、ＦｌｏｗＭａｒｋアプリケーション３４２'は、ＰＥＣにアクティビティをディスパッチし、ＦｌｏｗＭａｒｋデータベース４３８を更新して、アクティビティが走行中であることを示す。その後、ＰＥＣは、このプロセスの第１のアクティビティに登録されたアクティビティ・プログラム４３２を始動する。その後、選択されたアクティビティ・プログラム４３２が走行を開始する。ウェブ・クライアントの自動車レンタル要求を処理するのに必要なアクティビティ・プログラム４３２の数と性質は、プロセス・モデル４４０をどのように設計したかに依存する。たとえば、あるプロセス・モデル４４０では、アクティビティが、完了まで人間の介入なしに走行する完全に自動的なプロセスとしてモデル化される可能性がある。その代わりに、プロセス・モデル４４０で、モデル・プロセスを完了する前に広範囲の人間の入力及び介入が必要になる場合もある。設計の如何にかかわらず、各ワークフローのプロセス・モデル４４０によって、モデル化されたプロセスまたは手順を完了するために指定されたタスクを達成するのに必要な特定のアクティビティ・プログラム４３２が開始され

る。

【０１００】個々のアクティビティ・プログラム４３２のそれぞれは、特定のタスクを達成するか、要求された情報を返し、その後終了するように設計された、別々のソフトウェア・モジュールである。WWW　ＡＰＩ４３４を用いると、アクティビティ・プログラム４３２が、WWWを介してウェブ・クライアントとの間でデータ及び状況を送受信できるようになる。WWW　ＡＰＩ４３４は、必ずアクティビティ・プログラム４３２によって呼び出されるが、データ及び状況情報は、ＦＭＩＧ４３０とアクティビティ・プログラム４３２の間で両方向に流れる。

【０１０１】この例のレンタ・カー要求を処理するために、複数の関連するアクティビティ・プログラム４３２を呼び出して、プロセス・モデル４４０で指定されたワークフロー・プロセスを完了することができる。たとえば、１つのアクティビティ・プログラムを開始して、ＦｌｏｗＭａｒｋデータベース４３８に問い合わせ、レンタ・カーを要求している人物がそのレンタ・カー代理店の既存の顧客であるかどうかを判定することができる。そうである場合には、顧客識別番号が、ＦｌｏｗＭａｒｋデータベース４３８内の既存のアカウント番号と一致し、アクティビティ・プログラム４３２は、次に進んで自動車レンタル要求を処理できる。しかし、レンタ・カーを要求している人物が既存の顧客でない場合、異なるアクティビティ・プログラム４３２を開始して、ウェブ・クライアントと通信し、必要な情報を集め、ウェブ・クライアントの顧客情報をＦｌｏｗＭａｒｋデータベース４３８に入力することができる。

【０１０２】次に、有効な顧客識別が確立された後に、もう１つのアクティビティ・プログラム４３２を開始し、顧客の要求に従って要求を満足できるかどうかを判定することができる。これは、図２６のプロセス・ステップ２０２０に含まれる。たとえば、要求された日に、要求された都市に、要求されたサイズの使用可能な自動車があるかどうかを判定する。その代わりに、ＦｌｏｗＭａｒｋプロセス・モデルで、次の処理のために自動車レンタル要求を人間の代理人に経路指定することを指定することもできる。この場合、自動車レンタル要求は、代理人のＦｌｏｗＭａｒｋタスク・リストに表示される。その代わりに、このプロセス全体を完全に自動化することもできる。どの場合でも、レンタ・カー代理人またはアクティビティ・プログラム４３２は、ウェブ・クライアントの自動車レンタル要求を処理し、所望の自動車が使用可能な場合には、ＦｌｏｗＭａｒｋデータベース４３８を更新して、その自動車が「予約」済みであることを示す。これらのアクティビティは、図２６のプロセス・ステップ２０３０で達成される。しかし、所望の自動車が使用可能でない場合には、別のアクティビティ・プログラム４３２を開始して、別の位置で自動車を探し（図２６のプロセス・ステップ２０４０）、人間のレンタ・カー代理人から、所望の自動車を別の位置から所望の位置へ移動する要求を作成することができる（図２６のプロセス・ステップ２０５０）。これらのアクティビティ・プログラム４３２の一部を、下で詳細に説明する。

【０１０３】この例では、レンタ・カー代理人が、要求を是認し、車両を所望の位置に移動した後に、レンタ・カー代理人は、ＦｌｏｗＭａｒｋアプリケーション３４２'に入力を与え、アクティビティ・プログラム４３２を更新する。レンタ・カーがこのシステムで予約された後に、異なるアクティビティ・プログラム４３２を開始して、その顧客の確認番号を生成し、その顧客との確認のトランザクションを処理することができる。各アクティビティ・プログラム４３２は、最後まで実行した後に終了する、独立のプロセスとして設計される。

【０１０４】この時点で、アクティビティ・プログラム４３２は、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を使用して、ＦｌｏｗＭａｒｋ入力データ・コンテナから入力データを取り出す。ＦｌｏｗＭａｒｋデータ・コンテナは、プロセス・モデル４４０の作成時に定義されるＦｌｏｗＭａｒｋ機能である。ＦｌｏｗＭａｒｋデータ・コンテナは、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を介してアクセスでき、プロセス・モデル４４０内で状況及び情報を記憶し、あるアクティビティ・プログラム４３２から次のアクティビティ・プログラム４３２へ渡すための記憶位置として使用される。アクティビティ・プログラム４３２は、接続４２８を介してWWW　ＡＰＩ４３４にＯｐｅｎを発行することによって、ウェブ・クライアントとの会話をオープンする。アクティビティ・プログラム４３２には、実行中のプロセス・インスタンス、アクティビティ名、ＦｌｏｗＭａｒｋユーザＩＤなどの情報も含まれる。この情報は、WWW　ＡＰＩ４３４によってＦＭＩＧ４３０に送られる。ＦＭＩＧ４３０は、内部データ・キャッシュにある要求されたプロセス・インスタンス名を突き止めることによって、このＯｐｅｎ　ＡＰＩを適当なウェブ・クライアントと突き合わせる。その後、ＦＭＩＧ４３０は、このトランザクションの「会話識別子」を生成する。ＦＭＩＧ４３０は、その内部キャッシュに会話識別子を保管し、応答メッセージの一部としてWWW　ＡＰＩ４３４に会話識別子を送る。WWW　ＡＰＩ４３４は、ＦＭＩＧ４３０からの切離しを行い、会話識別子をアクティビティ・プログラム４３２に返す。この時点から、この会話識別子は、ウェブ・クライアントの要求を処理するのに必要な、このウェブ・クライアントとすべてのアクティビティ・プログラム４３２の間のWWW　ＡＰＩ４３４発行のすべてに含まれる。この会話識別子は、アクティビティ・プログラム４３２が発行するＣｌｏｓｅ　ＡＰＩによって会話が終了するまで有効である。

【０１０５】ＦＭＩＧ４３０から会話識別子を受け取った後に、アクティビティ・プログラム４３２は、WWW　ＡＰＩ４３４にＲｅｃｅｉｖｅを発行する。Ｒｅｃｅｉｖｅ　ＡＰＩは、ＦＭＩＧ４３０に対する、ウェブ・クライアントからデータを取得する要求である。この場合、ＦＭＩＧ４３０は、アクティビティ・プログラム４３２への発行を待っている、ウェブ・クライアントからのデータを有する。これは、このプロセスで前にＣＧＩ４２０がＦＭＩＧ４３０に送ったものと同一のウェブ・クライアント・データである。ＦＭＩＧ４３０は、このｐｏｓｔデータ及び環境データをWWW　ＡＰＩ４３４（Ｒｅｃｅｉｖｅ）に送り、Ｒｅｃｅｉｖｅ　ＡＰＩによって、そのデータがアクティビティ・プログラム４３２に中継される。アクティビティ・プログラム４３２は、そのデータを処理し、要求を満たすために適当なステップを実行する。

【０１０６】このプロセスのこの時点で、アクティビティ・プログラム４３２は、クライアント・ワークステーション２１０のウェブ・クライアントに予約確認ＨＴＭＬ画面を送信するＳｅｎｄをWWW　ＡＰＩ４３４に発行する。このＳｅｎｄ　ＡＰＩによって、ＦＭＩＧ４３０に対する送信要求が生成され、予約確認画面を描画するのに必要なＨＴＭＬデータが送られる。ＦＭＩＧ４３０は、要求に含まれる会話識別子とウェブ・クライアント・ハンドル（一緒に内部データ・キャッシュに記憶されている）を突合せ、まだＦＭＩＧ４３０に接続されて入力を待っているＣＧＩ４２０に、データ型及びハンドルと共にアクティビティ・プログラム４３２からのＨＴＭＬデータを送る。アクティビティ・プログラム４３２は、データ型「ＨＴＭＬテンプレート」を生成し、その結果、ＣＧＩ４２０が、元のＨＴＭＬテンプレートを解析でき、アクティビティ・プログラム４３２によって送られたＨＴＭＬデータを用いて適当なＨＴＭＬ置換変数を置換できるようにする。その代わりに、アクティビティ・プログラム４３２は、ＭＩＭＥ、ＵＲＬまたはＨＴＭＬテンプレート・データをＣＧＩ４２０に送ることができる。この時点で、ＣＧＩ４２０は、ＦＭＩＧ４３０からの切離しを行い、受け取ったデータの処理を開始する。その後、ＦＭＩＧ４３０は、応答メッセージ内のＯＫの戻りコードによってWWW　ＡＰＩ４３４に応答する。WWW　ＡＰＩ４３４も、ＦＭＩＧ４３０からの切離しを行い、アクティビティ・プログラム４３２への戻りコードを生成する。一般に、WWW　ＡＰＩ４３４は、各WWW　ＡＰＩ４３４が呼び出された後にＦＭＩＧ４３０からの切離しを行う。

【０１０７】予約確認テンプレートのＨＴＭＬコードの例を、図１３に示す。変数wf-act-outmsgは、ウェブ・ブラウザ２１２によって表示されるＨＴＭＬ画面内の確認番号によって置換される置換変数であることに留意されたい。ＦＭＩＧ４３０から受け取ったデータをＣＧＩ４２０が処理した後のＨＴＭＬコードを、図１４に示す。図１４のＨＴＭＬコードは、ウェブ・サーバ・アプリケーション２２２が変更するコードであり、ウェブ・ブラウザ２１２がクライアント・ワークステーション２１０に表示するコードである。図１４のＨＴＭＬコードには、顧客の予約番号が含まれることに留意されたい。この場合、ＣＧＩ４２０とウェブ・クライアントは、すでに会話を確立しており、ＣＧＩ４２０は、初めて呼び出された時からまだ活動状態であるから、ウェブ・クライアントを認証する必要はない。ＣＧＩ４２０は、隠された変数「wf-fmig-handle」を確認番号と共に挿入し、また、最初のＨＴＭＬ画面からのwf-fmig-keyをこのＨＴＭＬコードに挿入する。これによって、後続のＨＴＭＬページに、そのページからＦｌｏｗＭａｒｋ４５０へのアクセスを許可する埋込みデータを含めることができるようになる。この例に示されたＨＴＭＬ変数の置換は、ＪＡＶＡＳｃｒｉｐｔ変数の置換及び複製と、ＪＡＶＡＳｃｒｉｐｔテンプレートの解析及び挿入を含むように拡張することができる。本発明は、ＨＴＭＬ変数の置換に制限されない。

【０１０８】予約確認テンプレートの処理中に、アクティビティ・プログラム４３２は、WWW　ＡＰＩ４３４にＲｅｃｅｉｖｅ　ＡＰＩを発行しており、これが、顧客が確認番号を受信し、見たことを保証する、アクティビティ・プログラム４３２のための確認メッセージとして働く。ウェブ・クライアントは、自分の確認番号を受信し、記録した時に、「発行」ボタンをクリックする。このデータは、前と同様に、ウェブ・ブラウザ２１２によって保持されているウェブ・ユーザのユーザＩＤ及びパスワードと共に、ウェブ・サーバ・アプリケーション２２２に送信される。ウェブ・サーバ・アプリケーション２２２は、この情報を使用して、もう一度ＣＧＩ４２０に対してウェブ・クライアントを認証する。ウェブ・サーバ・アプリケーション２２２は、前と同一のＨＴＭＬコード、<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">を用いてＣＧＩ４２０を呼び出す。ＣＧＩ４２０は、ウェブ・ブラウザ２１２からのｐｏｓｔデータを受け取る。この例では、ウェブ・ブラウザ２１２は、データを「form-urlencoded media type」として書式化する。このデータは、図１５に示されたものに類似した外見になる。１３というwf-cgi-submitの値は、データをアクティビティ・プログラムに渡すコマンドである。ＦＭＩＧ４３０は、wf-fmig-keyを検索し、ｐｏｓｔデータからのキーと環境データからのウェブ・ユーザＩＤを検査して、ウェブ・クライアントが許可されることを確認する。ＦＭＩＧ４３０は、wf-fmig-handleを内部データ・キャッシュの会話識別子と突合せ、WWW　ＡＰＩ４３４が接続されているかどうかを検査する。WWW　ＡＰＩ４３４がまだ接続されていない場合、ＦＭＩＧ４３０は単に待機する。ＣＧＩ４２０は、ＦＭＩＧ

４３０に接続されたままになり、応答を待ち続ける。ＲｅｃｅｉｖｅのＷＷＷ　ＡＰＩ４３４がＦＭＩＧ４３０に接続された後に、ＦＭＩＧ４３０は、会話識別子を使用して、Ｒｅｃｅｉｖｅ　ＡＰＩを適当なウェブ・クライアントと突合せ、ｐｏｓｔデータと環境データをＷＷＷ　ＡＰＩ４３４に送り、ＷＷＷ　ＡＰＩ４３４は、このデータをアクティビティ・プログラム４３２に渡す。アクティビティ・プログラム４３２は、データを処理し、Ｃｌｏｓｅ　ＷＷＷ　ＡＰＩを使用して会話をクローズする。Ｃｌｏｓｅ　ＡＰＩは、会話識別子、プロセス・インスタンス名及び他の関連データと共にＦＭＩＧ４３０に送られる。ＦＭＩＧ４３０は、クローズ要求を処理し、応答メッセージをＷＷＷ　ＡＰＩ４３４に送り、ＷＷＷ　ＡＰＩ４３４は、会話が終了したことを検査するため、アクティビティ・プログラム４３２への戻りコードを生成する。

【０１０９】その後、ＦＭＩＧ４３０は、内部データ・キャッシュから記憶された変数値を取り出し、そのデータをＣＧＩ４２０に送る。ＣＧＩ４２０は、データと変数を受け取り、ＦＭＩＧ４３０からの切離しを行う。その後、ＣＧＩ４２０は、ウェブ・サーバ・アプリケーション２２２に連絡し、変数とデータを送り、ウェブ・ブラウザ２１２への指定されたＨＴＭＬを描画するようにウェブ・サーバ・アプリケーションに指示する。

【０１１０】この時点で、自動車レンタル要求に関するユーザとの対話は完了している。しかし、その間に完了している可能性がある、上で述べた他のアクティビティ・プログラム４３２がある。さらに、他のプロセスが、完了を必要とする可能性があり、これらのプロセスを達成するために他のアクティビティ・プログラム４３２が活動化される可能性がある。この例では、１つのアクティビティ・プログラム４３２が、ユーザと対話し、ユーザから集めたデータをＦｌｏｗＭａｒｋデータ・コンテナに送っていた。ＦｌｏｗＭａｒｋデータ・コンテナのデータは、現在は、プロセス・モデル４４０内の別のアクティビティ・プログラム４３２のための入力になっている。他のアクティビティ・プログラム４３２の一部を、下で説明する。

【０１１１】アクティビティ・プログラム４３２のいずれかが動作を完了した後に、制御がＰＥＣに戻される。ＰＥＣは、ＦｌｏｗＭａｒｋにリターンし、ＦｌｏｗＭａｒｋデータベース４３８が、１つのアクティビティ・プログラム４３２が完了したことと、このプロセスを完了するための次のアクティビティ・プログラム４３２を開始できることを示すように更新される。次のアクティビティ・プログラム４３２が、自動的に実行されるアクティビティ・プログラム４３２である場合、ＦｌｏｗＭａｒｋアプリケーション３４２'は、ＦｌｏｗＭａｒｋデータベース４３８内にアクティビティのインスタンスを作成し、ＰＥＣが活動状態であるかどうかを検査し、アクティビティをＰＥＣにディスパッチし、アクティビティが走行中であることを示すようにＦｌｏｗＭａｒｋデータベース４３８を更新する。ＰＥＣは、次に走行するアクティビティとして登録されているアクティビティを始動し、適当なアクティビティ・プログラム４３２が、完了するまで走行する。

【０１１２】上で注記したように、自動車レンタル要求を処理するのに必要なアクティビティの一部を、人間の対話または介入を必要とするようにモデル化することができる。たとえば、必要な場合には、自動車の予約要求を検査するために、人間のレンタ・カー代理人を必要とすることができる。その代わりに、要求された自動車が要求された位置で使用可能でない場合に、別の位置から適当な自動車を移動するために人間の代理人に連絡することができる。ウェブ・クライアントの自動車の要求の結果は、レンタ・カー代理店が要求された日に宛先の都市で使用可能な要求された自動車を有するか否かの２つがありえる。ＦｌｏｗＭａｒｋデータベースへの照会によって使用可能性を検査した後に、正しいアクティビティ・プログラムが開始される。

【０１１３】この例では、アクティビティ・プログラム４３２が、ＦｌｏｗＭａｒｋユーザのワーク・リストの項目を生成し、ＦｌｏｗＭａｒｋユーザ、この場合ではレンタ・カー代理人が、アクティビティを手動で始動しなければならない。レンタ・カー代理人は、ウェブ・ブラウザ２１２を使用して、ＦｌｏｗＭａｒｋ実行時クライアント・ウェブ・ページにアクセスする。レンタ・カー代理人用の適当なウェブ・ページを生成するのに必要なＨＴＭＬコードは、図１６に示されたコードに類似したものになるはずである。レンタ・カー代理人は、自分のキーを入力し、「Work with Work Items（ワーク項目に関する作業）」を選択し、発行ボタンをクリックする。このラジオボタンは、図１６のＨＴＭＬに<INPUT TYPE="radio" NAME="wf-cgi-html" VALUE="/exm/html/exmp5ewi.htm" CHECKED>という行があるので、特定のＨＴＭＬテンプレートすなわちexmp5ewi.htmに関連する。

【０１１４】やはり、前に述べたように、ウェブ・サーバ・アプリケーション２２２は、ウェブ・ブラウザ２１２から生成されたｐｏｓｔデータ・ストリームを受信する。ＣＧＩ４２０は保護されているので、もう１つのＲＥＡＬＭ要求を発行し、応答を得なければならない。前と同様に、このＲＥＡＬＭ要求では、レンタ・カー代理人がパスワードとユーザＩＤを入力することが必要である。ウェブ・ブラウザ２１２の将来の実施態様のいくつかでは、ウェブ・サーバ・アプリケーション２２２に送信されるデータ・ストリームにウェブ・ブラウザ２１２が何らかの形の認証データを挿入する機構が設けられる可能性が高いことに留意されたい。認証データを収集し、ウェブ・ブラウザ２１２からウェブ・サーバ・アプリケーション２２２に送信するための方法及び技法のす

べてが、本発明の範囲に含まれる。一般に、各ウェブ実行時クライアントは、ウェブ上でのユーザＩＤとパスワードを知っていなければならない。どの場合でも、ウェブ・ブラウザ２１２は、ウェブのユーザＩＤとパスワードを記憶し、この情報をウェブ・サーバ・アプリケーション２２２に送信する。前と同様に、ウェブ・サーバ・アプリケーション２２２は、ウェブ・ユーザＩＤ及びパスワードを使用して、ウェブ・クライアントを認証し、ＣＧＩ４２０へのアクセス権を得る。レンタ・カー代理人が認証された後に、ＣＧＩ４２０が、上の例のＨＴＭＬ言語、<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">を介して呼び出される。ＣＧＩ４２０は、ｓｔｄｉｎを介してウェブ・サーバ・アプリケーション２２２からｐｏｓｔデータ・ストリームを受け取る。このデータ・ストリームは、通常は図１７の例に似た外見になる。

【０１１５】データ・ストリームの発行（wf-cgisubmit）の値が０なので、ＣＧＩ４２０は、指定されたテンプレート（exmp5ewi.htm）を取り出し、ＨＴＭＬ変数についてテンプレートを解析し、解析された変数を、環境データ及び制御情報と共にＦＭＩＧ４３０に渡す。ＣＧＩ４２０が処理を開始する前にテンプレートを生成するのに使用されるＨＴＭＬコードの例は、図１８及び図１９に類似するはずである。

【０１１６】ＦＭＩＧ４３０は、データ・ストリームからレンタ・カー代理人のキー値（webfmagent）とウェブ・ユーザＩＤを取り出し、この情報を使用して、レンタ・カー代理人が使用を許可されているＦｌｏｗＭａｒｋユーザＩＤを、ＦｌｏｗＭａｒｋデータベース４３８から見つける。レンタ・カー代理人は、まだＦｌｏｗＭａｒｋ４５０にログ・インしていないので、ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋデータベース４３８内の相関するＦｌｏｗＭａｒｋ情報を使用してＦｌｏｗＭａｒｋ４５０にログ・インする。ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を使用して、ユーザによって前に入力された、自動車レンタル要求に関連する情報（すなわち、wf-api-item-id、wf-api-item-descrip、wf-api-item-stateなど）を取り出し、まだＦＭＩＧ４３０に接続され、ＦＭＩＧ４３０からのデータを待っているＣＧＩ４２０のためのメッセージにこのデータを書式化する。このデータは、ＦＭＩＧ４３０を介してＣＧＩ４２０に渡される。データがＣＧＩ４２０に送られた後に、ＣＧＩ４２０は、ＦＭＩＧ４３０からの切離しを行い、受け取ったデータの処理を開始する。ＣＧＩ４２０は、ＦＭＩＧ４３０からの情報を取り、変数を処理し、その情報を、クライアント・ワークステーション２１０に表示される適当なテンプレートに置く。ＣＧＩ４２０が処理を完了した後のＨＴＭＬコードの例を、図２０、図２１及び図２２に示す。図２０、図２１及び図２２に示されたＨＴＭＬコードは、クライアント・ワークステーション２１０上でレンタ・カー代理人用のワーク・リストのワーク項目の出力画面を描画するのに使用される。レンタ・カー代理人は、ＦｌｏｗＭａｒｋ４５０によって生成され、ウェブ・ブラウザ２１２によって表示されたワーク項目のリストを見る。レンタ・カー代理人は、表示されたワーク項目のうちの１つを選択し、「start work item（ワーク項目開始）」ボタンをクリックし、「perform action（アクション実行）」ボタンをクリックする。レンタ・カー代理人によるこれらの動作によって、ウェブ・ユーザＩＤ及びパスワードの情報を含む、ウェブ・サーバ・アプリケーション２２２へのデータ・ストリームが生成される。ウェブ・ユーザＩＤ及びパスワードは、ウェブ・サーバ・アプリケーション２２２によって保持され、レンタ・カー代理人がこの情報を再入力する必要はない。ウェブ・サーバ・アプリケーション２２２は、記憶したユーザＩＤ及びパスワードを使用して、レンタ・カー代理人を認証し、ＣＧＩ４２０へのアクセス権を得る。ウェブ・サーバ・アプリケーション２２２は、前に使用したものに類似したＨＴＭＬコード、<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">を用いてＣＧＩ４２０を呼び出し、ＣＧＩ４２０は、ウェブ・ブラウザ２１２からｐｏｓｔデータを受信する。ｐｏｓｔデータは、ｓｔｄｉｎを介してＣＧＩ４２０によって受信されるが、図２３に示されたデータに類似する。

【０１１７】この場合、レンタ・カー代理人がワーク項目開始の要求を発行しているので、submit変数の値は３に等しい。ＣＧＩ４２０は、指定されたテンプレート（この場合はexmp5ewi.htm）を検索し、ＨＴＭＬ変数についてテンプレートを解析し、これらの変数を、データ・ストリーム、環境データ及び制御情報と共にＦＭＩＧ４３０に渡す。この時点で制御はＦＭＩＧ４３０に渡されるが、ＣＧＩ４２０は、ＦＭＩＧ４３０に接続されたままで、ＦＭＩＧ４３０からデータが返されるのを待つ。ＦＭＩＧ４３０は、ＣＧＩ４２０が送った変数を使用して、レンタ・カー代理人を認証する。レンタ・カー代理人は、すでにＦｌｏｗＭａｒｋ４５０にログ・オンしているので、ＦＭＩＧ４３０は、レンタ・カー代理人をもう一度ログ・オンする必要はない。ＦＭＩＧ４３０は、データ・ストリームを解析し、ワーク項目（wf-api-item）開始の要求を見つけ、ワーク項目の処理を開始するためにＦｌｏｗＭａｒｋ　ＡＰＩ４３６を発行する。ワーク項目は、ＨＴＭＬ変数を用いてＷＷＷコンテキストで送信されているので、ＦＭＩＧ４３０は、アクティビティ・プログラムがＷＷＷ　ＡＰＩ４３４を使用してウェブ・クライアントとの会話を有することを知っている。ＦＭＩＧ４３０は、このウェブ・クライアントのハンドルを生成し、ハンドル、プロセス・インスタンス名、アクティビティ名、ユーザＩＤなどを内部キャッシュに入力する。また、ＦＭＩＧ４３０は、ＨＴＭＬ変

数情報を保管し、その結果、会話が行われる時にＨＴＭＬ変数を復元できるようにする。その後、ＦＭＩＧ４３０は、ＷＷＷ　ＡＰＩ４３４の接続を待つ。

【０１１８】要求されたワーク項目（wf-api-item）を開始するＦｌｏｗＭａｒｋ　ＡＰＩ４３６に応答して、ＦｌｏｗＭａｒｋは、ＦｌｏｗＭａｒｋデータベース４３８内にアクティビティのインスタンスを作成し、ＰＥＣが活動状態であるかどうかを検査し、アクティビティをＰＥＣにディスパッチし、ＦｌｏｗＭａｒｋデータベース４３８を更新してアクティビティが走行中であることを示す。ＰＥＣは、このプロセスで次のアクティビティに登録されているアクティビティ・プログラム４３２を始動する。アクティビティ・プログラムは、完了まで走行する。

【０１１９】アクティビティ・プログラム４３２は、ＦｌｏｗＭａｒｋ　ＡＰＩを使用して、ＦｌｏｗＭａｒｋ入力データ・コンテナから入力データを取り出す。アクティビティ・プログラムは、ＷＷＷ　ＡＰＩ４３４にＯｐｅｎ　ＡＰＩを発行することによってウェブ・クライアントとの会話をオープンする。Ｏｐｅｎ　ＡＰＩによって、プロセス・インスタンス、アクティビティ名、ＦｌｏｗＭａｒｋユーザＩＤなどの情報を含む、ＦＭＩＧ４３０用のＯｐｅｎ要求メッセージが生成される。ＦＭＩＧ４３０は、この情報を受け取り、内部データ・キャッシュ内の適当なプロセス・インスタンス、アクティビティ名及びユーザＩＤを見つけることによって、ウェブ・クライアントを照合する。前と同様に、ＦＭＩＧ４３０は、会話識別子も生成し、この識別子を内部キャッシュに記憶する。ＦＭＩＧ４３０は、応答メッセージでＷＷＷ　ＡＰＩ４３４に会話識別子を送り、ＷＷＷ　ＡＰＩ４３４は、ＦＭＩＧ４３０からの切離しを行い、会話識別子をアクティビティ・プログラム４３２に返す。この時点から、会話識別子が、このウェブ・クライアント及びこのプロセス・インスタンスを扱うすべてのＷＷＷ　ＡＰＩ４３４の発行に含まれるようになる。会話識別子は、この会話がクローズされるまで有効になる。アクティビティ・プログラム４３２は、ＷＷＷ　ＡＰＩ４３４にＳｅｎｄＡＰＩを発行して、ウェブ・クライアントへのＨＴＭＬ画面を生成する。アクティビティ・プログラム４３２は、データ型としてＨＴＭＬテンプレートを指定し、その結果、ＣＧＩ４２０に、テンプレートを解析し、アクティビティ・プログラム４３２によって送られたデータを用いて変数を置換することを知らせる。アクティビティ・プログラム４３２は、テンプレートの位置と、各変数の置換のためのテキストを指定する。

【０１２０】Ｓｅｎｄ　ＡＰＩによって、ＦＭＩＧ４３０に対する送信要求が生成され、データが送信される。ＦＭＩＧ４３０は、送信要求に含まれる会話識別子と適当なウェブ・クライアントを突合せ、アクティビティ・プログラム４３２からのデータ、データ型及びハンドルを、まだ接続されているＣＧＩ４２０に送る。この時点で、ＣＧＩ４２０は、ＦＭＩＧ４３０からの切離しを行い、受け取ったデータの処理を開始する。ＦＭＩＧ４３０は、ＷＷＷ　ＡＰＩ４３４に応答し、ＷＷＷＡＰＩ４３４も、ＦＭＩＧ４３０からの切離しを実行する。ＷＷＷ　ＡＰＩ４３４は、アクティビティ・プログラム４３２に戻りコードを送る。ＣＧＩ４２０は、ＦＭＩＧ４３０から送られたデータを使用して、顧客の要求を満たすために使用可能な自動車に関する情報を含む画面を作成する。さらに、ＣＧＩ４２０は、ウェブ・サーバ・アプリケーション２２２と通信しているので、ＣＧＩ４２０は、隠された変数「wf-fmig-handle」と、最初のＨＴＭＬ画面からのwf-fmigキーを、新しい画面のＨＴＭＬコードに挿入する。その後、ＣＧＩ４２０は、テンプレートを解析し、「wf-」変数を処理して、適当な位置に情報を書き込む。テンプレート・ファイルのＨＴＭＬコードは、図２４に示されたＨＴＭＬコードに類似した外見になる。ＣＧＩ４２０が処理を終了した後には、ファイル内のＨＴＭＬコードは、図２５のＨＴＭＬコードに類似した外見になる。これは、ウェブ・サーバ・アプリケーション２２２が、クライアント・ワークステーション２１０上のレンタ・カー代理人の画面を描画するのに使用されるＨＴＭＬコードである。その間に、アクティビティ・プログラム４３２は、ＲｅｃｅｉｖｅのＷＷＷＡＰＩ４３４を発行して、ウェブ・クライアントからデータを受信している。レンタ・カー代理人は、この自動車レンタル要求を満たすために予約する自動車を選択し、「submit process（プロセス発行）」ボタンをクリックする。ウェブ・ブラウザ２１２、ウェブ・サーバ・アプリケーション２２２、ＣＧＩ４２０、ＦＭＩＧ４３０及びＦｌｏｗＭａｒｋ４５０の間の通信処理は、上の説明と同様に行われる。最終的に、選択された自動車が、ＦｌｏｗＭａｒｋデータベース４３８内で更新され、会話が終了する。自動車レンタル予約プロセス・インスタンスは、この時点で完了するので、ＦｌｏｗＭａｒｋ４５０は、ＦｌｏｗＭａｒｋデータベース４３８からこのプロセス・インスタンスを除去する。

【０１２１】同様のイベントのシーケンスが、要求された位置に使用可能なレンタ・カーがない状況でも発生する。唯一の相違は、この場合、要求された都市に条件を満たす自動車がないので、レンタ・カー代理人の画面に、使用可能な自動車に関する情報が含まれないことである。この画面には、代替位置にある、顧客の要求に一致する自動車を移動するオプションが含まれるはずである。

【０１２２】その間に、アクティビティ・プログラム４３２は、Ｄｉｓｃｏｎｎｅｃｔ　ＡＰＩを発行している。Ｄｉｓｃｏｎｎｅｃｔ　ＡＰＩによって、会話識別子、プロセス・インスタンス、アクティビティ名、ＦｌｏｗＭａｒｋユーザＩＤなどの情報を含む、ＦＭＩＧ４

３０に関する切断要求メッセージが生成される。ＦＭＩＧ４３０は、この情報を受け取り、内部キャッシュ内の対応する項目を突き止め、この情報のすべてを、アクティビティ・プログラム４３２の「切断」状況と共に記録する。ＦＭＩＧ４３０は、ＷＷＷ　ＡＰＩ４３４にＯＫの応答メッセージを送り、ＷＷＷ　ＡＰＩ４３４は、ＦＭＩＧ４３０からの切離しを行い、アクティビティ・プログラム４３２に戻りコードを送る。アクティビティ・プログラム４３２は、戻りコードを検査し、切断ＡＰＩが成功裡に記録されたことを検査する。アクティビティ・プログラム４３２は、切断されたプロセスを後程復元するために、会話識別子と状況情報を局所データベースに保管する。このアクティビティは、実際に完了することなくできる限り進行しているので、制御はＰＥＣに戻される。アクティビティは完了していないので、ＰＥＣは、ＦｌｏｗＭａｒｋデータベース４３８を更新して、アクティビティが再始動の準備ができていることを示す。ＦｌｏｗＭａｒｋに関しては、これは「手動始動」アクティビティである。このアクティビティが、作動可能であり、手動始動であるので、ＦＭＩＧ４３０は、後程、レンタ・カー代理人からのデータが使用可能になった時に、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を発行して、アクティビティを再始動することができる。レンタ・カー代理人は、自動車レンタル要求を満たすために代替位置から自動車を移動することを選択し、「発行」をクリックする。データ転送とウェブ・クライアント認証の処理は、前に説明した形で進行する。この場合、ＦＭＩＧ４３０は、会話識別子を検査した時に、アクティビティの状況が「切断」であることを知る。ＦＭＩＧ４３０は、ＦｌｏｗＭａｒｋ　ＡＰＩ４３６を発行して、アクティビティ・プログラム４３２を再始動する。したがって、アクティビティ・プログラム４３２が再始動され、前と同様に、ＦｌｏｗＭａｒｋ４５０がプロセス・インスタンスを作成し、ＦｌｏｗＭａｒｋデータベース４３８を更新する。ＰＥＣが、アクティビティ・プログラム４３２を始動し、必要なデータが、ＦＭＩＧ４３０から転送される。この時点で、アクティビティ・プログラム４３２は、完了まで走行することができる。どのレンタ・カーをどの位置から移動するかを判定し、類似したシーケンス及び並びの通信によって、このデータをレンタ・カー代理人からＦｌｏｗＭａｒｋ４５０へ送ることが可能である。

【０１２３】上に示した例からわかるように、ＷＷＷ上でさまざまなソフトウェア・アプリケーションにアクセスするための共通ユーザ・インターフェースを提供することによって、高い効率と共に高い生産性が可能になる。ユーザがさまざまなソフトウェア・アプリケーションへのアクセス権を得るのに必要な訓練の長さと量の両方を減らしつつ、より多くのソフトウェア・アプリケーションを使用するための柔軟性も得られる。これらの長所のすべてが、競争力があり、市場で成功するビジネスを可能にする。

【０１２４】本発明の好ましい実施例に関して本発明を具体的に図示し、説明してきたが、当業者であれば、本発明の趣旨及び範囲から逸脱することなく、形態及び詳細におけるさまざまな変更を行うことができることを理解するであろう。

【０１２５】まとめとして、本発明の構成に関して以下の事項を開示する。

【０１２６】（１）少なくとも１つの中央処理装置（ＣＰＵ）と、ＣＰＵに結合されたメモリと、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、共通ユーザ・インターフェースを介して複数のウェブ・ブラウザへまたはこれらからデータを送受する能力を有し、データの識別及び追跡のために識別子機構を使用する、トランザクション・サポート機構とを含む、ワールド・ワイド・ウェブ上で複数のウェブ・ブラウザとソフトウェア・アプリケーションとの間で通信するための共通ユーザ・インターフェースを提供するコンピュータ・システム。

（２）さらに、セキュリティ機構を含み、セキュリティ機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、セキュリティ機構が、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間に結合され、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間のインターフェースを提供し、セキュリティ機構が、複数のウェブ・ブラウザからユーザ入力を受け取り、セキュリティ機構が、受け取った入力に対応するソフトウェア・アプリケーションの認証パラメータを取り出す、上記（１）のコンピュータ・システム。

（３）さらに、インターフェース機構を含み、インターフェース機構が、少なくとも１つの変数を処理するためのゲートウェイ機構を含み、ゲートウェイ機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ゲートウェイ機構が、ソフトウェア・アプリケーションの再プログラミングを必要としない、複数のウェブ・ブラウザとソフトウェア・アプリケーションとの間の通信のための汎用共通ゲートウェイ・インターフェースを含む、上記（１）のコンピュータ・システム。

（４）さらに、切断機構を含み、切断機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、切断機構が、ソフトウェア・アプリケーション・プロセスが再開される時にデータを取り出すことができるように、ソフトウェア・アプリケーション・プロセスが中止される時に、複数のウェブ・ブラウザのうちの１つとソフトウェア・アプリケーション・プロセスとの間の会話のそれぞれに関連する状態データ及び会話識別子を記憶する、上記（１）のコンピュータ・システム。

（５）メモリ内に常駐し、少なくとも１つのＣＰＵによ

って実行され、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間に結合され、ソフトウェア・アプリケーションと複数のウェブ・ブラウザとの間のインターフェースを提供し、複数のウェブ・ブラウザからユーザ入力を受け取り、受け取った入力に対応するソフトウェア・アプリケーションの認証パラメータを取り出す、セキュリティ機構と、少なくとも１つの変数を処理するためのゲートウェイ機構を含み、ゲートウェイ機構が、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ゲートウェイ機構が、ソフトウェア・アプリケーションの再プログラミングを必要としない、複数のウェブ・ブラウザとソフトウェア・アプリケーションとの間の通信のための汎用共通ゲートウェイ・インターフェースを含む、インターフェース機構と、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ソフトウェア・アプリケーション・プロセスが再開される時にデータを取り出すことができるように、ソフトウェア・アプリケーション・プロセスが中止される時に、複数のウェブ・ブラウザのうちの１つとソフトウェア・アプリケーション・プロセスとの間の会話のそれぞれに関連する状態データ及び会話識別子を記憶する、切断機構とをさらに含む、上記（１）のコンピュータ・システム。

（６）トランザクション・サポート機構が、さらに、ソフトウェア・アプリケーションへのネイティブ・インターフェースと通信するための機構を含む、上記（１）のコンピュータ・システム。

（７）トランザクション・サポート機構が、ウェブ・サーバ・アプリケーション及びソフトウェア・アプリケーションと通信するアプリケーション・ゲートウェイを含み、アプリケーション・ゲートウェイが、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行され、アプリケーション・ゲートウェイが、識別子機構を含み、識別子機構が、複数のウェブ・ブラウザのそれぞれのために識別子を生成し、ソフトウェア・アプリケーションからのデータを、複数のウェブ・ブラウザのうちの識別子に対応する選択された１つに経路指定する、上記（１）のコンピュータ・システム。

（８）アプリケーション・ゲートウェイが、複数のウェブ・サーバから受け取るデータを処理し、アプリケーション・プログラムから受け取るデータを処理する、上記（７）のコンピュータ・システム。

（９）ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、上記（１）のコンピュータ・システム。

（１０）さらに、ソフトウェア・アプリケーションの指示の下で実行される少なくとも１つのアクティビティ・プログラムと通信する少なくとも１つのアクティビティ・プログラム・インターフェース（ＡＰＩ）を含み、少なくとも１つのアクティビティ・プログラム・インターフェースが、少なくとも１つのアクティビティ・プログラムとアプリケーション・ゲートウェイとの間で通信する、上記（１）のコンピュータ・システム。

（１１）複数の中央処理装置（ＣＰＵ）と、複数のＣＰＵに結合されたメモリと、それぞれがメモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、複数のウェブ・ブラウザと、複数のウェブ・ブラウザのうちの少なくとも１つと通信する、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ウェブ・サーバ・アプリケーションと、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ソフトウェア・アプリケーションと、ウェブ・サーバ・アプリケーション及びソフトウェア・アプリケーションへのネイティブ・インターフェースと通信する、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、アプリケーション・ゲートウェイとを含み、アプリケーション・ゲートウェイが、複数のウェブ・ブラウザのそれぞれのために識別子を生成し、ソフトウェア・アプリケーションからのデータを、複数のウェブ・ブラウザのうちの識別子に対応する選択された１つに経路指定する、識別子機構を含むワールド・ワイド・ウェブ上でウェブ・ブラウザとソフトウェア・アプリケーションとの間で通信するための共通ユーザ・インターフェースを提供するコンピュータ・システム。

（１２）アプリケーション・ゲートウェイが、ウェブ・サーバ・アプリケーション及びアプリケーション・プログラムから受け取るデータを処理する、上記（１１）のコンピュータ・システム。

（１３）ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、上記（１１）のコンピュータ・システム。

（１４）さらに、ソフトウェア・アプリケーションの指示の下で実行される少なくとも１つのアクティビティ・プログラムと通信する少なくとも１つのアクティビティ・プログラム・インターフェース（ＡＰＩ）を含み、少なくとも１つのアクティビティ・プログラム・インターフェースが、少なくとも１つのアクティビティ・プログラムとアプリケーション・ゲートウェイとの間で通信する、上記（１１）のコンピュータ・システム。

（１５）ウェブ・サーバ・アプリケーションが、複数のウェブ・ブラウザのうちの１つから渡された認証データから、選択されたウェブ・ブラウザがウェブ・サーバ・アプリケーションへのアクセスを許可されるかどうかを判定する認証機構を含み、ウェブ・サーバが、複数のウェブ・ブラウザから受け取るデータ及びアプリケーション・ゲートウェイから受け取るデータを処理する上記（１１）のコンピュータ・システム。

（１６）ウェブ・ブラウザが、複数のＣＰＵのうちの少なくとも１つによってクライアント・ワークステーショ

ン上で実行される、上記（１１）のコンピュータ・システム。
（１７）ウェブ・サーバ・アプリケーションが、複数のＣＰＵのうちの少なくとも１つによって、ウェブ・サーバ・コンピュータ上で実行される、上記（１１）のコンピュータ・システム。
（１８）アプリケーション・ゲートウェイが、複数のＣＰＵのうちの少なくとも１つによって、ウェブ・サーバ・コンピュータ上で実行される、上記（１１）のコンピュータ・システム。
（１９）アプリケーション・ゲートウェイが、複数のＣＰＵのうちの少なくとも１つによって、第１のコンピュータ上で実行される、上記（１１）のコンピュータ・システム。
（２０）ソフトウェア・アプリケーションが、複数のＣＰＵのうちの少なくとも１つによって、第２のコンピュータ上で実行される、上記（１１）のコンピュータ・システム。
（２１）アプリケーション・ゲートウェイが、複数のＣＰＵのうちの少なくとも１つによって、第２のコンピュータ上で実行される、上記（１１）のコンピュータ・システム。
（２２）複数の中央処理装置（ＣＰＵ）を提供するステップと、複数のＣＰＵに結合されたメモリを提供するステップと、複数のＣＰＵのうちの少なくとも１つによって、メモリ内に常駐する複数のウェブ・ブラウザのうちの少なくとも１つを実行するステップと、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ウェブ・サーバ・アプリケーションを提供するステップと、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、ソフトウェア・アプリケーションを提供するステップと、メモリ内に常駐し、複数のＣＰＵのうちの少なくとも１つによって実行される、アプリケーション・ゲートウェイを提供するステップと、認証データ及び環境データをウェブ・サーバ・アプリケーションに送ることによって、複数のウェブ・ブラウザのうちの選択された１つが、ソフトウェア・アプリケーションへのアクセスを開始するステップと、認証データが選択されたウェブ・ブラウザにウェブ・サーバ・アプリケーションへのアクセスを許可する場合に、環境データを処理するステップと、処理された環境データをアプリケーション・ゲートウェイに出力するステップと、選択されたウェブ・ブラウザと、ソフトウェア・アプリケーションによって実行される所望のプロセスとに対応する識別子を生成するステップと、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ウェブ・ブラウザからユーザ入力を受け取り、受け取った入力に対応するソフトウェア・アプリケーションの認証パラメータを取り出す、セキュリティ機構を提供するステップと、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、ウェブ・ブラウザとソフトウェア・アプリケーションとの間で変数及びテンプレートを送受する、インターフェース機構を提供するステップと、メモリ内に常駐し、少なくとも１つのＣＰＵによって実行され、会話が再開されてソフトウェア・アプリケーションによって所望のプロセスが実行される時に状態データを取り出すことができるように、会話が中止される時にウェブ・ブラウザとソフトウェア・アプリケーションとの間の会話に関連する状態及び会話識別子を記憶する、切断機構を提供するステップと、所望のプロセスを実行した結果を、識別子を有するアプリケーション・ゲートウェイに返すステップと、識別子に基づいて、複数のブラウザのうちのどれに結果を送らなければならないかを決定するステップと、アプリケーション・ゲートウェイからウェブ・サーバ・アプリケーションに結果を送るステップと、識別子に対応する選択された１つのウェブ・ブラウザに、サーバから結果を送るステップとを含む、ワールド・ワイド・ウェブ上でウェブ・ブラウザとソフトウェア・アプリケーションとの間で通信するための共通ユーザ・インターフェースを提供するための、コンピュータ実施される方法。
（２３）ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、上記（２２）の方法。
（２４）ウェブ・ブラウザを走行させるクライアント・ワークステーションと、ウェブ・サーバ・アプリケーションを走行させるウェブ・サーバ・コンピュータと、アプリケーション・ゲートウェイを走行させる第１コンピュータと、ソフトウェア・アプリケーションを走行させる第２コンピュータと、ウェブ・ブラウザとウェブ・サーバ・アプリケーションとの間でデータを伝送できるようにする、ウェブ・ブラウザとウェブ・サーバ・アプリケーションとの間の通信機構と、ウェブ・サーバ・アプリケーションとアプリケーション・ゲートウェイとの間でデータを伝送できるようにする、ウェブ・サーバ・アプリケーションとアプリケーション・ゲートウェイとの間の通信機構と、アプリケーション・ゲートウェイとソフトウェア・アプリケーションとの間でデータを伝送できるようにする、アプリケーション・ゲートウェイとソフトウェア・アプリケーションとの間の通信機構と、ウェブ・ブラウザとソフトウェア・アプリケーションとの間で変数及びテンプレートを送受する、インターフェース機構と、ウェブ・ブラウザとソフトウェア・アプリケーションとの間に結合され、ウェブ・ブラウザとソフトウェア・アプリケーションとの間のインターフェースを提供する、セキュリティ機構と、会話が再開されてソフトウェア・アプリケーションによって所望のプロセスが実行される時に状態データを取り出すことができるように、会話が中止される時にウェブ・ブラウザとソフトウェア・アプリケーションとの間の会話に関連する状態及

び会話識別子を記憶する、切断機構と、ウェブ・ブラウザがワールド・ワイド・ウェブ上でソフトウェア・アプリケーションと通信できるようにする、複数のアプリケーション・プログラミング・インターフェースとを含む、ワールド・ワイド・ウェブ上でウェブ・ブラウザとソフトウェア・アプリケーションの間で通信するための共通ユーザ・インターフェースを提供するためのシステム。

（２５）ソフトウェア・アプリケーションが、プロセス・エンジニアリング・ソフトウェア・アプリケーションである、上記（２４）のシステム。

（２６）ウェブ・サーバ・コンピュータが、第１コンピュータを含む、上記（２４）のシステム。

（２７）第１コンピュータが、第２コンピュータを含む、上記（２４）のシステム。

【図面の簡単な説明】

【図１】本発明の好ましい実施例のブロック図である。

【図２】クライアント・ワークステーションとウェブ・サーバの間のトランザクションを示すブロック図である。

【図３】標準ウェブ・ブラウザからワールド・ワイド・ウェブを介してソフトウェア・アプリケーションにアクセスできるようにする、本発明の好ましい実施例によるシステムのブロック図である。

【図４】ワールド・ワイド・ウェブを介するＦｌｏｗＭａｒｋワークフロー・アプリケーション・ソフトウェアへのアクセスにさらに適合された、図３のシステムの詳細なブロック図である。

【図５】共通ユーザ・インターフェースの特徴の一部を示す、本発明の好ましい実施例による方法の流れ図である。

【図６】図５のセキュリティ検査機能の処理流れ図である。

【図７】図５のテンプレート／ＨＴＭＬ変数機能の処理流れ図である。

【図８】図５のＲＥＳＵＭＥ　ＡＰＩ機能及びＤＩＳＣＯＮＮＥＣＴ　ＡＰＩ機能の処理流れ図である。

【図９】ユーザ・ライブラリの一部を表す表である。

【図１０】複数ユーザー環境に拡張された時の本発明の好ましい実施例のブロック図である。

【図１１】本発明の好ましい実施例による、自動車レンタル予約フォームの生成に使用されるＨＴＭＬコードの例を示す図である。

【図１２】本発明の好ましい実施例に従って、ウェブ・ブラウザからのユーザ要求によって生成されるデータ・ストリームの例を示す図である。

【図１３】本発明の好ましい実施例による、予約確認テンプレートの生成に使用されるＨＴＭＬコードの例を示す図である。

【図１４】本発明の好ましい実施例に従って、ＣＧＩによって処理された後の図１０のＨＴＭＬコードの例を示す図である。

【図１５】本発明の好ましい実施例に従ってウェブ・ブラウザが書式化したデータの例を示す図である。

【図１６】本発明の好ましい実施例による、自動車レンタル予約代理人用のウェブ・ページの生成に使用されるＨＴＭＬコードの例を示す図である。

【図１７】本発明の好ましい実施例に従ってウェブ・サーバが生成したデータの例を示す図である。

【図１８】本発明の好ましい実施例による、レンタ・カー代理人作業リストの生成に使用されるＨＴＭＬコードの例を示す図である。

【図１９】本発明の好ましい実施例による、レンタ・カー代理人作業リストの生成に使用されるＨＴＭＬコードの例を示す図である。

【図２０】本発明の好ましい実施例に従ってＣＧＩによって処理された後の図１５のＨＴＭＬコードの例を示す図である。

【図２１】本発明の好ましい実施例に従ってＣＧＩによって処理された後の図１５のＨＴＭＬコードの例を示す図である。

【図２２】本発明の好ましい実施例に従ってＣＧＩによって処理された後の図１５のＨＴＭＬコードの例を示す図である。

【図２３】本発明の好ましい実施例による、ウェブ・ブラウザからＣＩＧによって受信されたデータ・ストリームの例を示す図である。

【図２４】本発明の好ましい実施例に従って自動車使用可能性ページを生成するのに使用されるＨＴＭＬコードの例を示す図である。

【図２５】本発明の好ましい実施例に従ってＣＧＩによって処理された後の図１８のＨＴＭＬコードの例を示す図である。

【図２６】本発明の好ましい実施例を使用するソフトウェア・アプリケーションとのＷＷＷトランザクションを記述した、処理モデル図である。

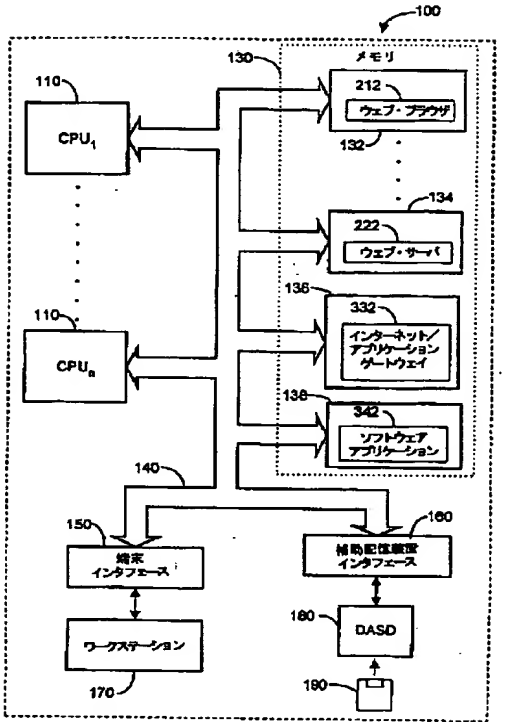
【符号の説明】

１００　システム
１１０　中央処理装置（ＣＰＵ）
１３０　メモリ
１３２　記憶位置
１３４　記憶位置
１３６　記憶位置
１３８　記憶位置
１４０　バス
１５０　端末インターフェース
１６０　補助記憶装置インターフェース
１７０　ワークステーション
１８０　直接アクセス記憶装置（ＤＡＳＤ）
１９０　フロッピ・ディスク

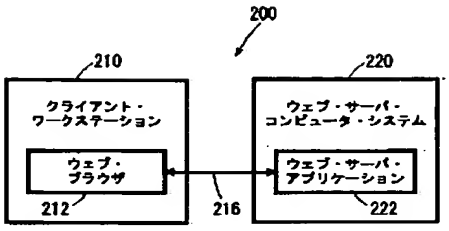
63　　64

２１２　ウェブ・ブラウザ　　＊ェイ（ゲートウェイ）
２２２　ウェブ・サーバ・アプリケーション　　３４２　ソフトウェア・アプリケーション
３３２　インターネット／アプリケーション・ゲートウ＊

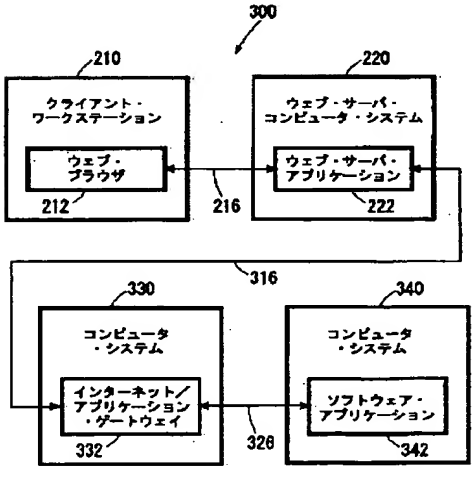
【図１】



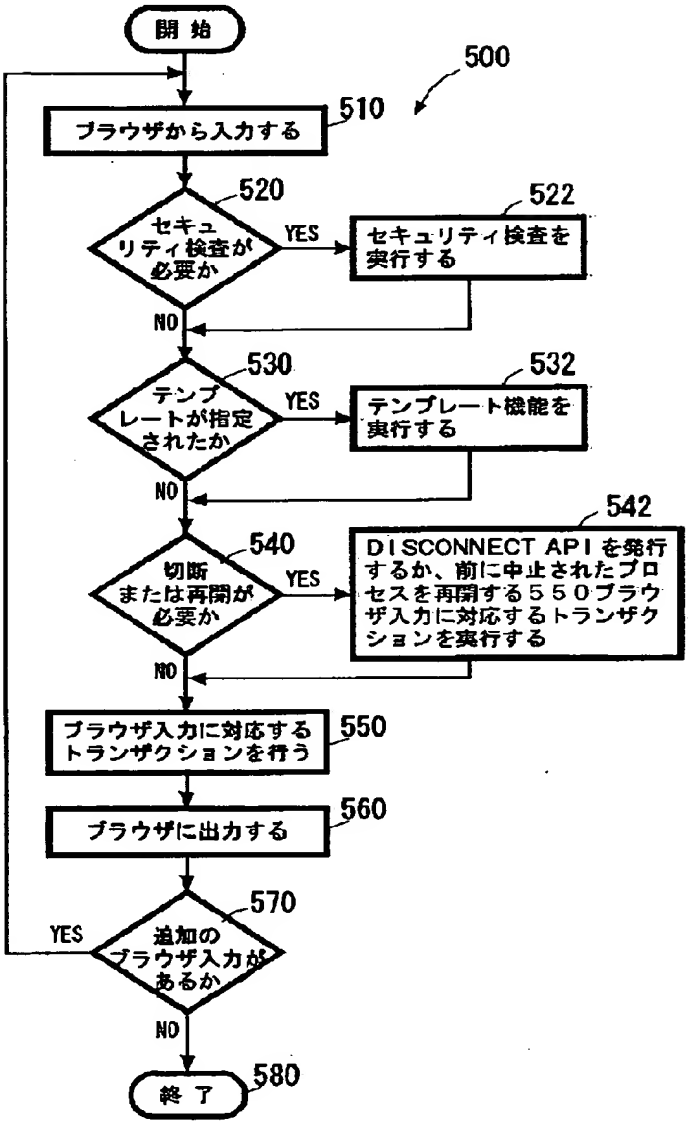
【図２】



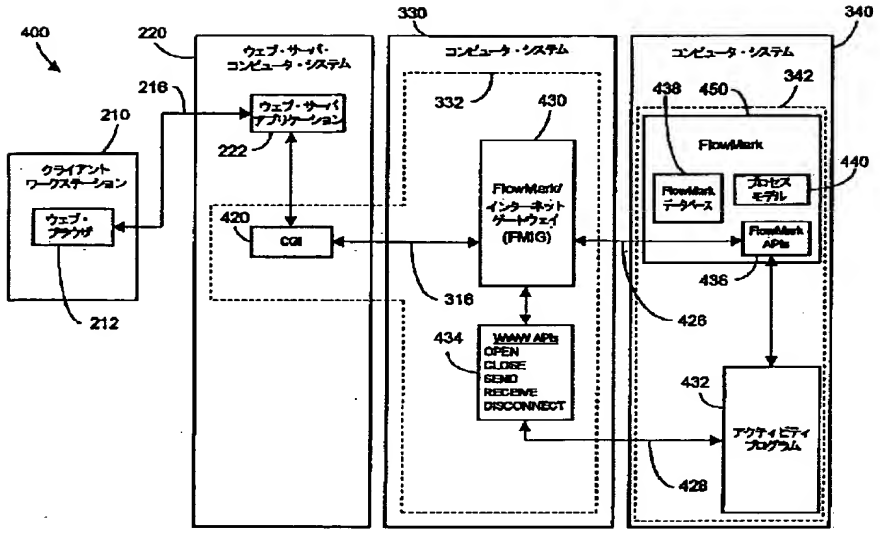
【図３】



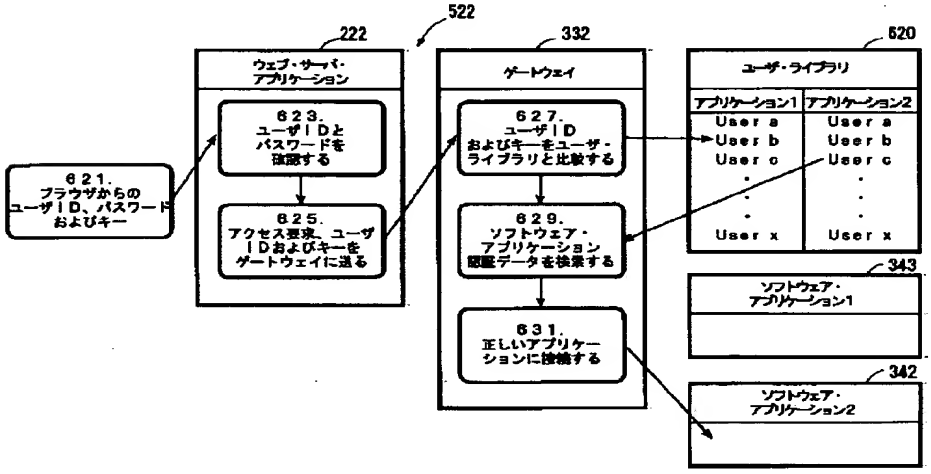
【図５】



【図４】



【図６】



【図９】

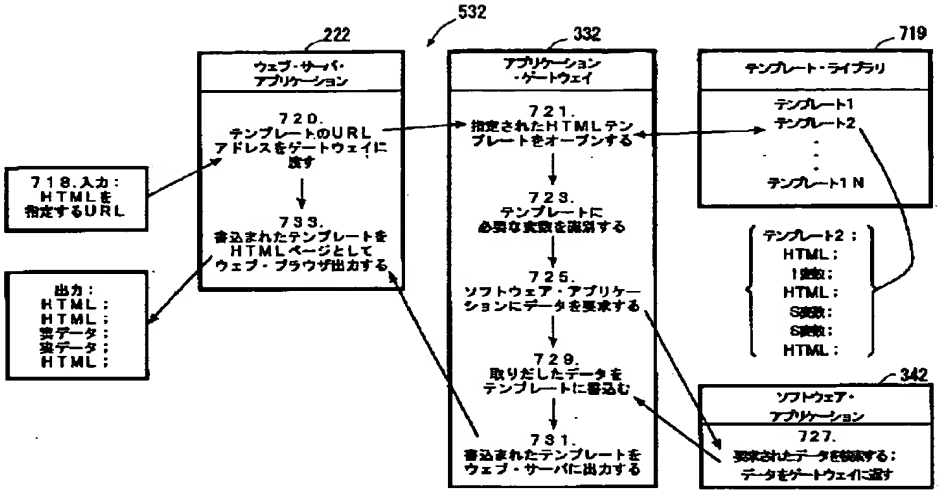
900

データベース・サーバ	ユーザ名	パスワード	データベース・サーバ	データベース
User a	Joe Brown	Secret	CorpServer	CorpInfo
User b				
User c				
User x				

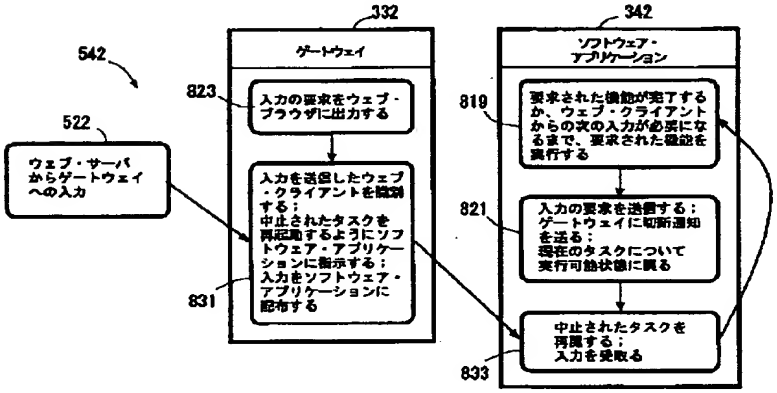
【図１３】

```
<html>
<title>Cars Around the World</title>
<FORM ACTION="/cgi-pro/tsemp5cgi.exe" METHOD="POST">
<H3><CENTER>Customer's Reservation Number</CENTER></H3>
<HR>
<p> <! -- "wf-act-outmsg" -->
<HR>
<p><INPUT TYPE="submit" VALUE="OK">
<p><INPUT TYPE="hidden" NAME="wf-cgi-submit" VALUE="13">
</form>
</html>
```

【図７】



【図８】



【図１２】

```
membno=1234&lname=Doe&fname=John&mi=E&origcity=Nevada&origstate=IA&startdate=08%2F23%2F96&days=4&cartype=2&wf-cgi-submit=2&wf-api-proc-template=www_Reservation_Request&wf-fmig-key=webfmcust&wf-cgi-html=%2Fexm%2Fsmp%2Fexmp5srk.htm
```

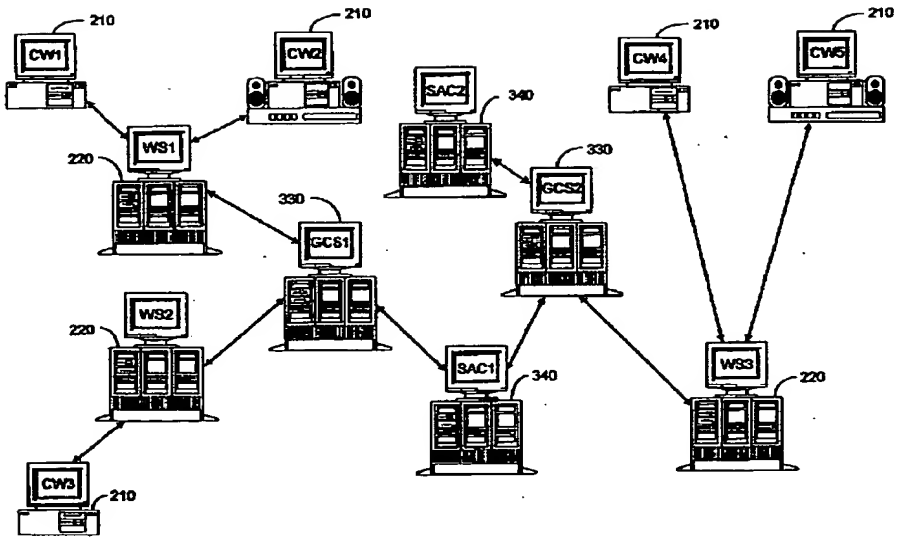
【図１５】

```
wf-cgi-submit=13&wf-fmig-handle=007777775F52657365727681746968wf-fmig-key=webfmcust
```

【図１７】

```
wf-fmig-key=webfmagent&wf-cgi-html=%2Fexm%2Fhtml%2Fexmp5ewi-htm&wf-cgisubmit=0
```

【図１０】



【図１１】

```
<html>
<title>Cars Around the World</title>
<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">
<H3>Rental Reservation</H3>
<HR>
Enter the following information and then press submit:
<HR>
<p>Member number:    <INPUT TYPE="text" NAME="membno" SIZE=4>
<p>Name:      Last:    <INPUT TYPE="text" NAME="lname" SIZE=15>
                      First:   <INPUT TYPE="text" NAME="fname" SIZE=15>
                      MI:      <INPUT TYPE="text" NAME="mi" SIZE=1>
<HR>
<P>Origin City:  <INPUT TYPE="text" NAME="origcity" SIZE=15>
        State:   <INPUT TYPE="text" NAME="origstate" SIZE=2>
<P>Start Date (MM/DD.YY): <INPUT TYPE="text" NAME="startdate" SIZE=8>
        Number of days <INPUT TYPE="text" NAME="days" SIZE=3>
<P>Select Car Preference:
        <INPUT TYPE="radio" NAME="cartype" VALUE=1> Compact
        <INPUT TYPE="radio" NAME="cartype" CHECKED VALUE=2> Mid Size
        <INPUT TYPE="radio" NAME="cartype" VALUE=3> Full Size
        <INPUT TYPE="radio" NAME="cartype" VALUE=4> Luxury
<HR>
<P><INPUT TYPE="submit" VALUE="Submit">
<P><INPUT TYPE="hidden" NAME="wf-cgi-submit" VALUE="2">
<P><INPUT TYPE="hidden" NAME="wf-api-proc-template" VALUE="www_Reservation Request">
<INPUT TYPE="hidden" NAME="wf-fmig-key" VALUE="webfmcust">
<INPUT TYPE="hidden" NAME="wf-cgi-html" VALUE="/exm/smp/exmp5srk.htm">
</form>
</html>
```

【図２３】

```
wf-api-item="7777775F5363686564756C655F52657365727661746966E007777775F52657365
72766174696F6E5F526571756573745F383500454D4147454E5400776562167656E7400776562
666D6167656E74"&wf-cgi--submit=3&wf-cgi-html=%2Fexm%2Fhtml%2Fexmp5ewi.htm&wf-fmig-
key=webfmagent
```

【図１４】

```
<html>
<title>Cars Around the World<title/>
<body>
<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">
<H3><CENTER>Customer's Reservation Number</CENTER></H3>
<HR>
<p> Your reservation number is 4411. </p>
<HR>
<p><INPUT TYPE="submit" VALUE="OK">
<p><INPUT TYPE="hidden" NAME="wf-cgi-submit" VALUE="13">
<INPUT TYPE="hidden" NAME="wf-fmig-handle" VALUE="007777775F52657365727661746966F">
<INPUT TYPE="hidden" NAME="wf-fmig-key" VALUE="webfmcust">
</form>
</html>
```

【図１６】

```
<html>
<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">
<HEAD>
<title>IBM Internet Connection for FlowMark home page</title>
</HEAD>
<BODY>
<h3>
IBM Internet Connection for FlowMark home page
</h3>
<hr>
<h4>
Enter key, select next screen, then press the submit button.
</h4>
<strong>
Key:
</strong>
<INPUT TYPE="text" NAME="wf-fmig-key" SIZE=64>
<br>
<INPUT TYPE="radio" NAME="wf-cgi-html" VALUE="/exm/html/exmp5ewi.htm" CHECKED>
<IMG SRC="/exm/icons/exmp5ewi.gif" ALIGN=MIDDLE HSPACE=5>Work with Work Items
<br>
<INPUT TYPE="radio" NAME="wf-cgi-html" VALUE="/exm/html/exmp5epi.htm">
<IMG SRC="/exm/icons/exmp5epi.gif" ALIGN=MIDDLE HSPACE=5>Work with Process Instances
<br>
<INPUT TYPE="radio" NAME="wf-cgi-html" VALUE="/exm/html/exmp5ept.htm">
<IMG SRC="/exm/icons/exmp5ept.gif" ALIGN=MIDDLE HSPACE=5>Work with Process Templates

<! - - Start Submit Button & hidden variables - - - - - - - - - - - - - - - >

<hr>
<INPUT TYPE="submit" VALUE="Submit">
<INPUT TYPE="hidden" NAME="wf-cgi-submit" VALUE="0">

<! - - End Submit Button & hidden variables - - - - - - - - - - - - - - - >
<br>
<b>
[
<a href="/exm/docs/exmp5d20.htm">Help |
<a href=http://www.ibm.com/">IBM home page</a>> |
<a href=http://www.software.ibm.com/ad/flowmark/exmn0mst.htm">FlowMark home page</a>
]

</FORM>
</BODY>
```

【図１８】

```
<html>
<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">
<HEAD>
<title>FlowMark - Work Items</title>
</HEAD>
<BODY>
<img src="/exm/icons/exmp5eww.gif" alt=Work with Work Items>
<h4>Select a work item:</h4>
<TABLE BORDER=1>
<tr>
        <th COLSPAN=2>Description
        <th>Status
        <th>Activity name
        <th>Process
        <th>Program
        <th>Received<br>date & time
        <th>Priority
        <th>Category
<! - - "wf-cgi-rbegin" - - >
<tr>
        <td ALIGN=CENTER VALIGN=MIDDLE><INPUT TYPE="radio"
                NAME="wf-api-item" VALUE="wf-api-item-id">
        <td> NOWRAP><! - - "wf-api-item-descrip" - - >
        <td><! - - "wf-api-item-state" - - >
        <td><! - - "wf-api-item-name" - - >
        <td><! - - "wf-api-item-procinst" - - >
        <td><! - - "wf-api-item-impl" - - >
        <td ALIGN=CENTER><! - - "wf-api-item-starttime" - - >
        <td ALIGN=CENTER><! - - "wf-api-item-priority" - - >
        <td><! - - "wf-api-item-category" - - >
<! - - "wf-cgi-rend" - - >
</TABLE>

<hr>
<h4>Next, select which action you want to perform and
<br>
press the button below.
</h4>

<TABLE BORDER=0>
<tr>
<td>
<INPUT TYPE="radio" NAME="wf-cgi-submit" VALUE="3" CHECKED>
Start work item
<tr>
<td>
```

【図１９】

```
<INPUT TYPE="radio" NAME="wf-cgi-submit" VALUE="0">
Refresh list
</TABLE>
<! - - Start Submit Button & hidden variables - - - - - - - - - - - - - - - ->
<hr>
<INPUT TYPE="submit" VALUE="Perform Action">
<INPUT TYPE="hidden" NAME="wf-cgi-html" VALUE="/exm/html/exmp5ewl.htm">
<! - - End Submit Button & hidden variables - - - - - - - - - - - - - - - ->
<hr>
<aref="/exm/html/exmp5ehp.htm">
<img src="/exm/icons/exmp5efm.gif" ALIGN=MIDDLE HSPACE=5>
Return to home page
</a>
<aref="/exm/html/exmp5d20.htm">
<img src="/exm/icons/exmp5eoh.gif" ALIGN=MIDDLE HSPACE=5>
Help
</a>
<hr>
<TABLE BORDER=1>
<tr>
<td>FlowMark User
<td>FlowMark Database
<td>FlowMark Server
<td>Date & Time
<tr>
<td><! - - "wf-api-fmuser" - - >
<td><! - - "wf-api-fmdb" - - >
<td><! - - "wf-api-fmserver" - - >
<td><! - - "wf-api-datetime" - - >
</table>
</FORM>
</BODY>
</HTML>
```

【図２２】

```
<TABLE BORDER=1>
<tr>
<td>FlowMark User
<td>FlowMark Database
<td>FlowMark Server
<td>Date & Time
<tr>
<td>FMAGENT
<td>WWWDB
<td>WWWSRV
<td>08:27:33 09/05/96
</table>
<INPUT TYPE="hidden" NAME="wf-fmig-key" VAULE="webfmagent">
</FORM>
</BODY>
</HTML>
```

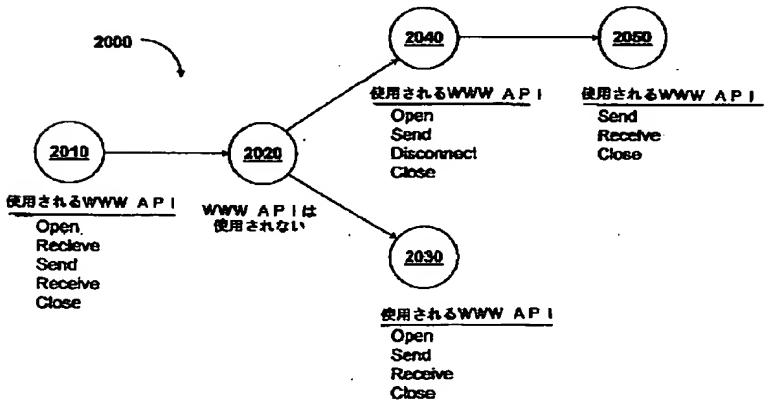
【図２０】

```
<html>
<FONT SIZE=3>
<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">
<HEAD>
<title>FlowMark - Work Items</title>
</HEAD>
<BODY>
<img src="/exm/icons/exmp5eww.gif" alt=Work with Work Items>
<h4>Select a work item:</h4>
<TABLE BORDER=1>
<tr>
        <th COLSPAN=2>Description
        <th>Status
        <th>Activity name
        <th>Process
        <th>Program
        <th>Received<br>date & time
        <th>Priority
        <th>Category
<! - - "wf-cgi-rbegin" - - >
<tr>
        <td ALIGN=CENTER VALIGN=MIDDLE><INPUT TYPE="radio"
NAME="wf-api-item"
VALUE="7777775F5363686564756C655F52657365727661746966F8E007777775F52657365727661
74696F8E5F52657175657374573745F383500454D4147454E5400776562616765E7400776562666D616
7656E74">
        <td> NOWRAP>Schedule car
        <td>Ready
        <td>www_Schedule_Reservation
        <td>www_Reservation Request_85
        <td>Schedule_Reservation
        <td ALIGN=CENTER>09-03-1996 02:55:11PM
        <td ALIGN=CENTER>4004128
        <td>
<tr>
        <td>ALIGN=CENTER VALIGN=MIDDLE><INPUT TYPE="radio"
NAME="wf-api-item"
VALUE="7777775F5363686564756C655F52657365727661746966F8E007777775F52657365727661
74696F8E5F52657175657374573745F383500454D4147454E5400776562616765E7400776562666D616
7656E74">
        <td> NOWRAP>Schedule car
        <td>Ready
        <td>www_Schedule_Reservation
        <td>www_Reservation Request_84
        <td>Schedule_Reservation
        <td ALIGN=CENTER>09-03-1996 01:48:48PM
        <td ALIGN=CENTER>4004128
        <td>
<tr>
        <td>ALIGN=CENTER VALIGN=MIDDLE><INPUT TYPE="radio"
```

【図２１】

```
NAME="wf-api-item"
VALUE="7777775F5363686564758C655F52657365727661746966F6E007777775F52657365727661
74696F6E5F52657175657375F383500454D4147454E540077858261676568E7400776582686D616
7656E74">
        <td> NOWRAP>Schedule car.
        <td>Ready
        <td>www_Schedule_Reservation
        <td>www_Reservation Request_86
        <td>Schedule_Reservation
        <td ALIGN=CENTER>09-03-1996 03:01:15PM
        <td ALIGN=CENTER>4004128
        <td>
<! - - "wf-cgi-rend" - - >
</TABLE>
<hr>
<h4>Next, select which action you want to perform and
<br>
press the button below.
</h4>
<TABLE BORDER=0>
<tr>
<td>
<INPUT TYPE="radio" NAME="wf-cgi-submit" VALUE="3" CHECKED>
Start work item
<tr>
<td>
<INPUT TYPE="radio" NAME="wf-cgi-submit" VALUE="0">
Refresh list
</TABLE>
<! - - Start Submit Button & hidden variables - - - - - - - - - - - - - - - - >
<hr>
<INPUT TYPE="submit" VALUE="Perform Action">
<INPUT TYPE="hidden" NAME="wf-cgi-html" VALUE="/exm/html/exmp5ewi.htm">
<! - - End Submit Button & hidden variables - - - - - - - - - - - - - - - - - >
<hr>
<ahref="/exn/html/exmp5ehp.htm">
<img src="/exn/icons/exmp5efm.gif" ALIGN=MIDDLE HSPACE=5>
Return to home page
</a>
<ahref="/exn/html/exmp5d20.htm">
<img src="/exm/icons/exmp5eoh.gif" ALIGN=MIDDLE HSPACE=5>
Help
</a>
<hr>
```

【図２６】



【図２４】

```
<HTML>
<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">
<HEAD>
<TITLE>Customer Information</TITLE>
<B>Customer Reservation Information</B>
<table border=1>
<tr><td>Name</td><td>Member Number</td><td>Resv Number</td>
<tr><td ><!--"wf-act-name"--></td>
<td align=right>"wf-act-membno"</td>
<td align=right>"wf-act-resvno"</td>
</table>
<p><!--"wf-act-outmsg"-->
<HR>
<B>Move An Available Ca</B>
<P><INPUT TYPE="radio" NAME=cars CHECKED> <!--"wf-act-car1"-->
<P><INPUT TYPE="radio" NAME=cars > <!--"wf-act-car2"-->
<P><INPUT TYPE="radio" NAME=cars > <!--"wf-act-car3"-->
<P><INPUT TYPE="radio" NAME=cars > <!--"wf-act-car4"-->
<HR>
<INPUT TYPE="submit" NAME="move" VALUE="Move Car">
<INPUT TYPE="submit" NAME="cancel" VALUE="Cancel">
<INPUT TYPE="hidden" NAME="wf-cgi-submit" VALUE="13">
</FORM>
</HTML>
```

【図２５】

```
<HTML>
<FORM ACTION="/cgi-prot/exmp5cgi.exe" METHOD="POST">
<HEAD>
<TITLE>Customer Information</TITLE>
<B>Customer Reservation Information</B>
<table border=1>
<tr><td>Name</td><td>Member Number</td><td>Resv Number</td>
<tr><td >John E. Doe</td>
<td align=right>1234</td>
<td align=right>4412</td>
</table>
<p>Reservation Date: <B>09/24/96</B> Days Requested:
<B>4</B><P>Car Type Requested: <B>Luxury</B><P>Origin: <B>Nevada, IA</b>
<HR>
<B>Move An Available Ca</B>
<P><INPUT TYPE="radio" NAME=cars CHECKED> XYD123 Pontiac Grand Am
<P><INPUT TYPE="radio" NAME=cars >Chevy Camaro Z28
<P><INPUT TYPE="radio" NAME=cars >Chevrolet Lumina
<P><INPUT TYPE="radio" NAME=cars >Oldmobile Cutlass Supreme
<HR>
<INPUT TYPE="submit" NAME="move" VALUE="Move Car">
<INPUT TYPE="submit" NAME="cancel" VALUE="Cancel">
<INPUT TYPE="hidden" NAME="wf-cgi-submit" VALUE="13">
<INPUT TYPE="hidden" NAME="wf-fmig-handle"
VALUE="7777775F4361725F556E617661696C61626C65007777775F526573657276617469696F6E5F
526571756573745F3837">
<INPUT TYPE="hidden" NAME="wf-fmig-key" VALUE="webmagent">
</FORM>
</HTML>
```

フロントページの続き

(72)発明者　ケニス・エドガー・ブラウン
アメリカ合衆国55901　ミネソタ州ロチェスターサーティーサード・ストリート　ノースウェスト　937
(72)発明者　バーネル・ジェームズ・ダイクス
アメリカ合衆国55920　ミネソタ州バイロン　サード・アベニュー　ノースウェスト　720
(72)発明者　エリック・ドゥエーン・リンドバーグ
アメリカ合衆国55906　ミネソタ州ロチェスターリバーサイド・レーン　ノースイースト　2685
(72)発明者　ダイアン・イレイン・オルソン
アメリカ合衆国55901　ミネソタ州ロチェスターナインティーンス・アベニュー　ノースウェスト　3910　ナンバー17
(72)発明者　ジェフリー・エドワード・セルデン
アメリカ合衆国32250　フロリダ州ジャクソンヴィル・ビーチ　ベンマン・ロード　920
(72)発明者　デヴォン・ダニエル・スナイダー
アメリカ合衆国55906　ミネソタ州ロチェスターグレンデール・ヒルズ・ドライブ　ノースイースト　1201
(72)発明者　ジェームズ・オリン・ワルツ
アメリカ合衆国55901　ミネソタ州ロチェスターナインティーンス・ストリート　ノースウェスト　408